



ESCOLA NAVAL

talant de bi-faire



Tiago Vieira Rodrigues

Instrumentação do Projeto VENT-SUP-EN

**Estudo e Projeto do Sistema de Telecomunicações e Sensores para um Veículo
Autónomo**

**Dissertação para obtenção do grau de Mestre em Ciências Militares Navais, na especialidade de
Engenheiros Navais – Ramo de Armas e Eletrónica**



Alfeite

2019



Tiago Vieira Rodrigues

Instrumentação do Projeto VENT-SUP-EN

***Estudo e Projeto do Sistema de Telecomunicações e Sensores para um Veículo
Autónomo***

Dissertação para a obtenção do grau de Mestre em Ciências Militares Navais, na especialidade
de Engenheiros Navais – Ramo de Armas e Eletrónica

Orientação de: Professor Doutor Bruno Duarte Damas

Coorientação de: Professor Doutor Custódio José de Oliveira Peixeiro

O Aluno Mestrando

O Orientador

Tiago Vieira Rodrigues
ASPOF EN-AEL

Professor Doutor Bruno Duarte
Damas

**Alfeite
2019**

“The only limit to our realization of tomorrow will be our doubts of today.”

Franklin D. Roosevelt

Agradecimentos

O desenvolver de qualquer trabalho reside não só no autor, mas também em todos aqueles que, de algum modo, contribuíram para a sua realização.

Aos meus orientadores, Prof. Bruno Damas e Prof. Custódio Peixeiro, por todos os conhecimentos transmitidos, por toda a disponibilidade, apoio e dedicação.

Um agradecimento especial a todos que tornaram possível a realização do presente trabalho: ao meu pai, avô e Sr. Aires pela ajuda durante a construção do monopolo clássico; aos técnicos Carlos Brito e António Almeida pelo auxílio na elaboração e testes da antena impressa; ao pessoal do CNOCA pela disponibilidade e auxílio durante os testes efetuados.

À minha namorada, pela compreensão nos muitos momentos de ausência, pela motivação, carinho e apoio nos momentos mais difíceis.

À minha família e amigos, por todo o apoio e incentivo nas diferentes etapas da minha vida. Em especial aos meus pais, por todos os valores e educação que me transmitiram.

Aos meus camaradas do Curso Jorge Álvares pelo contributo na minha formação académica e enriquecimento a nível pessoal ao longo destes anos de Escola Naval.

A todos que, direta ou indiretamente, contribuíram para esta dissertação e para a minha formação profissional e pessoal, gostaria de manifestar o meu profundo agradecimento.

Resumo

Mudanças climáticas, anormalidades ambientais, escassez de recursos humanos e assuntos de segurança nacional são problemáticas que têm pressionado as comunidades científicas, civis e militares para a investigação e estudo de USV (*Unmanned Surface Vehicle*). VENT-SUP-EN é um projeto do CINAV que visa a construção de um Veículo Não Tripulado de Superfície, capaz de operar em qualquer ambiente marítimo, como resposta aos problemas atuais.

Como contributo para o projeto detalhado deste veículo, o trabalho foi conduzido tendo por base três objetivos: fazer um estudo inicial dos sensores que podem equipar o veículo a ser construído; construir e operacionalizar o sistema de sensores básico para navegação; fazer o projeto, fabrico e teste das antenas do sistema de telecomunicações do veículo.

Numa primeira fase, foi feito um estudo sobre os sensores que deverão equipar o veículo, com recurso a uma abordagem teórica, seguindo-se a escolha desses sensores e finalmente a integração dos sensores básicos de navegação (*GPS* e *IMU*) no sistema.

Para o sistema de telecomunicações foram dimensionadas e otimizadas antenas, de forma a construir um sistema adaptado às especificações impostas. Construíram-se duas antenas: uma na banda dos 433 MHz (monopolo de fios) e outra na banda dos 2,4 GHz (monopolo impresso). Tendo em conta os aspetos teóricos, as antenas foram dimensionadas e otimizadas utilizando programas de simulação numérica, construídas em laboratório, e medidas as suas impedâncias de entrada. Para o protótipo de 2,4 GHz foi ainda medido o diagrama de radiação em câmara anecoica.

Como prova de conceito foram integrados os sistemas desenvolvidos, tendo-se realizado testes à capacidade do sistema, com recurso a uma plataforma que simula o mastro do veículo colocado numa embarcação, à medida que esta se afasta da estação em terra. Foram medidos vários parâmetros para que as valências do sistema fossem avaliadas consoante a distância entre a estação em terra e o navio.

Palavras-chave: VENT-SUP-EN, Sistema de Telecomunicações, Instrumentação, Antenas monopolo.

Abstract

Climate changes, environmental never before seen phenomenon, resource scarcity, and national security issues are problems that have pressured the civil and military scientific communities to investigate and study USV (*Unmanned Surface Vehicle*). VENT-SUP-EN is a CINAV project that aims to build a maritime Surface Unmanned Vehicle in response to those current problems.

As a contribution to the detailed design of this vehicle, the work was conducted based on three main goals: to carry out an initial study of the sensors that can gear up the vehicle; to build and test the basic navigation sensor system; to design, fabricate and test all of the vehicle's telecommunication system antennas.

On a first phase, a study was made about which sensors should be built into the vehicle, using a theoretical approach, followed by the choice of these sensors and, finally, the integration of basic navigation sensors (GPS and IMU) in the system.

For the telecommunications system, antennas were designed and optimized, in order to build a customized system fulfilling the specifications. Two antennas were built: one in the 433 MHz band (wire monopole) and one in the 2.4 GHz band (printed monopole). Taking into account the theoretical aspects, they were designed using simulation software tools, fabricated in a laboratory, and input impedance measured. Moreover, for the 2,4 GHz prototype, the radiation pattern was measured in an anechoic chamber.

As a proof of concept, the systems developed so far have been integrated, and the system capacity tests were carried out using a platform that simulates the mast of the vehicle placed on a boat as it moves away from the station at shore. Several parameters were measured so that the system's performance could be evaluated according to the distance between the station at shore and the ship.

Keywords: VENT-SUP-EN, Telecommunications System, Instrumentation, Monopole antennas.

Índice

Epígrafe	IV
Agradecimentos	VI
Resumo	VIII
Abstract.....	X
Índice	XII
Índice de Figuras	XVI
Índice de Tabelas	XVIII
Índice de Gráficos.....	XX
Lista de Abreviaturas, siglas e acrónimos	XXII
1 Introdução	1
2 Sensores de um Veículo Não Tripulado	2
2.1 Introdução	2
2.2 Tipos de Sensores	4
2.2.1 Módulo de Navegação	4
2.2.1.1 GPS.....	4
2.2.1.2 IMU	9
2.2.1.3 RADAR.....	11
2.2.1.4 Sonda	12
2.2.2 Módulo Visual	13
2.2.2.1 Câmara EO/IR.....	13
2.2.2.2 Sistema LIDAR.....	15
2.2.3 Módulo Subaquático	16
2.2.3.1 SONAR	16
2.2.3.2 Sensor de Poluição	18
2.2.4 Observações Meteorológicas	18
2.3 Interfaces de comunicação.....	19
2.3.1 RS-232	19
2.3.2 RS-422	19
2.3.3 RS-485	19
2.3.4 Serial Peripheral Interface	20
2.3.5 Inter-Integrated Circuit	20
2.3.6 NMEA.....	20

2.4	Trabalhos Relacionados	21
2.5	Escolha dos Sensores	23
2.5.1	GPS	23
2.5.2	IMU	24
2.5.3	RADAR	26
2.5.4	Câmara EO/IR	27
2.5.5	Sistema <i>LIDAR</i>	28
2.5.6	Sistema Subaquático	30
2.6	Conclusão	31
3	Sistema de Telecomunicações	32
3.1	Introdução	32
3.2	Definição dos Objetivos	33
3.3	Monopolo de Fios	36
3.3.1	Dimensionamento e Otimização	36
3.3.1.1	Monopolo Ideal	36
3.3.1.2	Plano de Terra	41
3.3.1.3	Monopolo Otimizado	45
3.3.2	Construção e Testes	46
3.4	Monopolo Impresso	49
3.4.1	Dimensionamento e Otimização	51
3.4.2	Construção e Teste	56
3.5	Conclusões	61
4	Integração dos Sistemas	64
4.1	Conceito	65
4.2	Microcontrolador - <i>ESP32 – Wemos TTGO t-beam</i>	67
4.3	Arquitetura do Protótipo	69
4.4	Protocolo RS232	70
4.5	Ligação <i>Ethernet</i>	72
4.6	Programação: Rotinas e Tarefas	73
4.7	Conclusões	74
5	Testes e Análise de Resultados	76
5.1	Banda 433 MHz	77

5.1.1	Arquitetura da Plataforma de Testes.....	77
5.1.2	Análise de Resultados.....	79
5.2	Banda 2,4 GHz.....	80
5.2.1	Arquitetura da Plataforma de Testes.....	81
5.2.2	Análise de Resultados.....	82
6	Conclusão.....	86
	Referências Bibliográficas.....	90
	Apêndice A – Código Protocolo RS232.....	96
	Apêndice B – Código <i>ESP32</i>	106
	Apêndice C – Código <i>Raspberry Pi</i>	114
	Apêndice D – Plataforma de Testes (433 MHz).....	118
	Apêndice E - Plataforma de Testes (433 MHz – Estação em Terra).....	120
	Apêndice F - Plataforma de Testes (433 MHz – VENT-SUP-EN).....	122
	Apêndice G - Plataforma de Testes (2.4 GHz – VENT-SUP-EN).....	126

Índice de Figuras

Figura 2.1: Funcionamento de uma estação <i>DGPS</i>	6
Figura 2.2: Constituição da rede <i>DGPS</i> Portuguesa.....	7
Figura 2.3: Funcionamento de um sistema <i>SBAS</i> (<i>WAAS</i>)	8
Figura 2.4: Diferentes sistemas <i>SBAS</i> consoante a região do utilizador	8
Figura 2.5: Funcionamento do sistema <i>RTK</i> e rede <i>ReNEP</i> Portuguesa	9
Figura 2.6: <i>IMU</i> composto por 3 sensores.....	10
Figura 2.7: Distância a um alvo num Radar de Impulsos.....	11
Figura 2.8: Diagrama de blocos de um sistema radar	12
Figura 2.10: Constituição de uma câmara IR (espectro infravermelho)	14
Figura 2.11: Determinação da distância ao objeto – <i>LIDAR</i>	15
Figura 2.12: Mapa de alta resolução.....	16
Figura 2.13: Funcionamento básico de um Sonar ativo	17
Figura 2.14: Imagem obtida por um sonar de multifeixe (cima) vs sonar lateral (baixo)	18
Figura 3.1: Esquema básico de um sistema de radiocomunicações	33
Figura 3.2: Monopolo vertical sobre um plano de terra infinito	35
Figura 3.3: Parâmetros obtidos na primeira simulação (condições iniciais)	37
Figura 3.4: Diagrama de radiação e geometria do monopolo (condições iniciais)	37
Figura 3.5: Obtenção da largura de banda através da variação da ROE em frequência. 41	
Figura 3.6: Diagramas de radiação de uma antena com plano de terra infinito: condutor perfeito (à esquerda) e água salgada (à direita) ($f = 433,92$ MHz).....	42
Figura 3.7: Simulação de um plano de terra electromagneticamente infinito	43
Figura 3.8: Monopolo com plano de terra constituído por 4 refletores e respetivo diagrama de radiação	43
Figura 3.9: Esquema de construção e respetivo resultado (monopolo de fios)	47
Figura 3.10: Torno utilizado para moldagem do monopolo	47
Figura 3.11: Medição do coeficiente de reflexão	48
Figura 3.12: Configuração de uma antena impressa.....	49

Figura 3.13: Tipos de alimentação de uma antena impressa	50
Figura 3.14: Medidas iniciais utilizadas no software <i>CST</i>	52
Figura 3.15: Monopolo impresso e conector SMA projetado no software <i>CST</i>	53
Figura 3.16: Sistema de eixos (XYZ) e ângulos de <i>Theta</i> e <i>Phi</i> usados.....	55
Figura 3.17: Diagramas de radiação do monopolo impresso ($f=2,45$ GHz).....	56
Figura 3.18: Máquinas utilizadas no processo (decapagem à esquerda e UV à direita) 56	
Figura 3.19: Máscara utilizada no processo fotolitográfico	57
Figura 3.20: Monopolo impresso de 2,45 GHz com conector SMA.....	57
Figura 3.21: Montagem para a medição do diagrama de radiação da antena.....	59
Figura 3.22: Diagramas de Radiação (Simulado vs Experimental).....	60
Figura 3.23: Arquitetura do sistema de telecomunicações	61
Figura 4.1: Arquitetura do Sistema de Controlo Remoto de um Veículo	64
Figura 4.2: Esquema básico do sistema a implementar.....	66
Figura 4.3: Influência dos parâmetros LoRa no alcance	68
Figura 4.4: Arquitetura do sistema C3 do VENT-SUP-EN.....	70
Figura 4.5: Protocolo <i>RS232</i> e mensagens possíveis.....	71
Figura 4.6: Mensagem formatada do protocolo <i>RS232</i>	72
Figura 4.7: Arquitetura alternativa do protótipo VENT-SUP-EN.....	73
Figura 5.1: Plataforma móvel de testes	76
Figura 5.2: Estação em Terra e Zona de <i>Fresnel</i>	77
Figura 5.3: Arquitetura da plataforma de testes (banda 433 MHz)	78
Figura 5.4: Dados de telemetria observados na estação em terra.....	79
Figura 5.5: Localização da estação em terra e percurso da embarcação	80
Figura 5.6: Arquitetura da plataforma de testes (banda 2,4 GHz).....	81
Figura 5.7: Página <i>Web</i> de gestão de ficheiros na estação em terra	82

Índice de Tabelas

Tabela 2.1: Estado da Arte, VENT-SUP existentes	22
Tabela 2.2: Tabela Comparativa GPS	23
Tabela 2.3: Tabela Comparativa IMU	25
Tabela 2.4: Tabela Comparativa RADAR.....	26
Tabela 2.5: Tabela Comparativa Câmara EO/IR	28
Tabela 2.6: Tabela Comparativa Sistema LIDAR.....	29
Tabela 2.7: Tabela Comparativa Sistemas Subaquáticos	31
Tabela 3.1: Otimização dos parâmetros do monopolo ideal variando o raio e altura	40
Tabela 3.2: Estudo do plano de terra ($f = 433,92$ MHz).....	45
Tabela 3.3: Quadro resumo dos resultados obtidos	46
Tabela 3.4: Otimização dos parâmetros do monopolo impresso.....	54
Tabela 4.1: Material utilizado no protótipo do VENT-SUP-EN	75
Tabela 5.1: Material utilizado na plataforma de testes (banda 433MHz)	78
Tabela 5.2: Material utilizado na plataforma de testes (banda 2.4 GHz)	81

Índice de Gráficos

Gráfico 3.1: Comportamento da relação de onda estacionária com a variação do número de segmentos	38
Gráfico 3.2: Comportamento do ganho com a variação do número de segmentos	38
Gráfico 3.3: Comportamento da impedância com a variação do número de segmentos	39
Gráfico 3.4: Curva do fator de reflexão do monopolo clássico	49
Gráfico 3.5: Parâmetros $S_{1,1}$ obtidos	55
Gráfico 3.6: Primeira versão do Coeficiente de reflexão ($S_{1,1}$) do monopolo impresso	58
Gráfico 3.7: Versão final do Coeficiente de reflexão ($S_{1,1}$) do monopolo impresso	58
Gráfico 3.8: Curvas do ganho nos diferentes planos das medições	60
Gráfico 5.1: Relação sinal ruído no router	83
Gráfico 5.2: Nível de Sinal Recebido	83
Gráfico 5.3: Taxa de transferência (<i>download</i>)	84
Gráfico 5.4: Tempos de latência	85

Lista de Abreviaturas, siglas e acrónimos

ADC *Analog to Digital converter*

ASV *Autonomous Surface Vehicle*

bps *bits per second*

C3 *Comando, Controlo e Comunicações*

CPU *Central Processing Unit*

CST *Computer Simulation Technology*

DAC *Digital to Analog Converter*

EGNOS *European Geostationary Navigation Overlay Service*

FSK *Frequency-shift Keying*

GFSK *Gaussian Frequency-shift Keying*

GMSK *Gaussian Minimum-shift Keying*

GPS *Global Positioning System*

IDE *Integrated Development Environment*

I²C *Inter-Integrated Circuit*

IMU *Inertial Measurement Unit*

IoT *Internet of Things*

ISM *Industrial, Scientific and Medical*

ITU-R *International Telecommunication Union – Radiocommunication Sector*

kbps	<i>kilobits per second</i>
LIDAR	<i>Light Detection and Ranging</i>
LoRa	<i>Long Range</i>
MSAS	<i>Multi-functional Satellite Augmentation System</i>
MSK	<i>Minimum-shift Keying</i>
NMEA	<i>National Marine Electronics Association</i>
NRBQ	Nuclear Radiológica Biológica e Química
OOK	<i>On-off Keying</i>
PWM	<i>Pulse Width Modulation</i>
RADAR	<i>Radio Detection and Ranging</i>
ROE	Relação de Onda Estacionária
RTK	<i>Real Time Kinematics</i>
SBAS	<i>Satellite-based Augmentation System</i>
SNR	<i>Signal to Noise Ratio</i>
SPI	<i>Serial Peripheral Interface</i>
SRAM	<i>Static Random-Access Memory</i>
UART	<i>Universal Asynchronous Receiver Transmitter</i>
USV	<i>Unmanned Surface Vehicle</i>
VENT-SUP-EN	Veículo Não Tripulado de Superfície
WASS	<i>Wide Area Augmentation System</i>

1 Introdução

Aproximadamente dois terços da superfície do planeta terra é oceano (Yuh, Marani, & Richard Blidberg, 2011), mas grande parte dessa área encontra-se inexplorada. Mudanças climáticas, anormalidades ambientais, escassez de recursos humanos e assuntos de segurança nacional são problemáticas que têm pressionado as comunidades científicas civis e militares para a investigação e estudo de Veículos Não Tripulados de Superfície (VENT-SUP), conhecidos como *USV's (Unmanned Surface Vehicles)* ou *ASV's (Autonomous Surface Vehicles)* (Liu, Zhang, Yu, & Yuan, 2016). Mas o que são estes veículos? Os veículos de superfície são plataformas robóticas que podem ser facilmente manuseadas por um único operador, com o objetivo de desempenhar missões de interesse a quilômetros de distância, entenda-se como superfície qualquer ambiente aquático, seja rio, mar, lago ou oceano, (Bertram, 2008).

O uso de plataformas não tripuladas para apoio a operações militares é um dos assuntos mais estudados na atualidade, onde a tecnologia substitui o Homem de forma a cumprir as missões de maior risco, poupando assim recursos e a própria vida humana.

Recorrendo a notas históricas, verificamos que o uso destas plataformas remonta ao século XVI. Em 1718, o Capitão Woodes Rogers¹ bloqueou as duas saídas do porto de Nassau² com uma força de dois navios de guerra da *Royal Navy* para impedir a saída dos piratas existentes e oferecer-lhes perdão. Para desfazer este bloqueio e fugir com os bens roubados, o Capitão Charles Vane³ usou uma técnica já utilizada pelos ingleses contra a Armada Espanhola em 1588, incendiando um dos seus navios e fazendo-o embater com os navios inimigos, sem guarnição a bordo (Woodard, 2008).

¹Woodes Rogers (1679-1732) foi o primeiro Governador Real das Bahamas, tendo sido também capitão e corsário inglês.

²Capital da Comunidade das Bahamas.

³Charles Vane (1680-1720) foi um dos mais temidos piratas do século XVIII que atuavam nas Caraíbas.

Nos últimos 15 anos, foram desenvolvidos e publicados vários estudos sobre UUVs (*Unmanned Underwater Vehicle*), enquanto os USVs (*Unmanned Surface Vehicle*) se mantiveram em segundo plano, apesar de não serem uma descoberta recente e tendo sido usados desde a Segunda Guerra Mundial (Corfield & Young, 2006). Várias missões foram desempenhadas por USVs desenvolvidos na época, tais como a recolha de amostras de água radioativa após testes de rebentamento de bombas atómicas (*Operation Crossroads*), em operações de dragagem de minas no Vietname e em exercícios de tiro, onde eram usados USVs como alvos (Graham, 2008).

Portugal possui uma vasta área marítima sob jurisdição nacional e é o quinto país da Europa com maior área, vigésimo a nível mundial, o que nos torna um país com enorme responsabilidade no que concerne o conhecimento, a vigilância, segurança e salvamento nesta área. Com os escassos recursos humanos, torna-se essencial criar estratégias que promovam a eficácia e a eficiência no desempenho das missões, deixando a tecnologia auxiliar na realização destas.

Os VENT-SUP (Veículo Não Tripulado de Superfície) são alvo de pesquisa e desenvolvimento por parte de várias nações, em particular da NATO⁴. Tornando-se assim necessário criar uma doutrina que regule os principais requisitos que estes veículos têm de cumprir, de forma a antecipar a resposta das nações a possíveis situações adversas. Um dos mais importantes requisitos definidos são as missões que estes veículos deverão desempenhar (Estado Maior da Armada, 2016):

- Operações anfíbias;
- Operações de busca e salvamento marítimo;
- Operações de guerra antiaérea;
- Operações de guerra antissubmarina;
- Operações de guerra de minas;
- Operações de guerra de superfície;
- Operações de guerra eletrónica;
- Operações de monitorização de infraestruturas;

⁴ *North Atlantic Treaty Organization*

- Operações de proteção de força e proteção portuária;
- Operações de recolha de dados ambientais;
- Operações de vigilância, fiscalização e segurança marítima;

VENT-SUP-EN é um projeto que visa a construção de um veículo de superfície não tripulado com requisitos operacionais bem definidos para a execução de tarefas que vão de encontro às necessidades da Marinha Portuguesa. Essas tarefas privilegiam a segurança, vigilância, salvamento, proteção ambiental e aquisição de dados.

A construção de um navio, autónomo ou não, tem de passar por várias fases de projeto. Segundo Lérias, o projeto de um navio passa por quatro fases: projeto conceptual, projeto básico, projeto detalhado e projeto de construção. O projeto VENT-SUP-EN irá também passar por estas quatro fases. Concluídas as fases de projeto conceptual e básico, é tempo de nos debruçarmos sobre o projeto detalhado, fase muito trabalhosa e extensa, que inclui o estudo de, por exemplo, o sistema de produção e conservação de energia, sistema de propulsão e manobra, o sistema integrado de comunicações, o sistema de navegação, sistemas de monitorização e recolha de dados ambientais, etc (Lérias, 2005).

O projeto conceptual foi desenvolvido no ano 2018 em que foram definidos requisitos para uma embarcação de casco planante que rondava os 8 metros. Mais tarde, em 2019, foram definidos diferentes requisitos com vista a construção de uma plataforma com as seguintes diretivas:

- Catamaran constituído por dois cascos modulares de espuma poliuretano e uma plataforma central onde estão agregados os equipamentos, feita de material rígido (polímero ou alumínio);
- Disposição modular que permita multimissão através da troca da plataforma central;
- Comunicação com a estação de controlo a uma distância não inferior a 5 milhas náuticas;
- Irá ter um desempenho operacional em condições de mar de “pequena vaga” (estado mar 3 na escala Douglas – até 1,25m de altura de vaga) e condições de vento “moderado” (4 na escala de Beaufort);

- Destina-se a ser operado por 2 (dois) homens em estação remota, podendo este ser transportado por meios terrestres ou navais;
- Velocidade máxima igual ou superior a 10 nós em mar calmo e vento até força 2 (dois) da escala de Beaufort;
- Autonomia igual ou superior a 1 (uma) hora à velocidade máxima;
- As principais dimensões do VENT-SUP-EN são aproximadamente as seguintes: comprimento fora-a-fora entre 2,5 a 3,2 metros; boca máxima igual ou superior a 2,5 metros; calado máximo igual ou inferior a 0,3 metros; deslocamento carregado igual ou inferior a 140 kg.

Esta dissertação irá incidir sobre o projeto detalhado do veículo, dando assim mais alguns contributos para a conclusão deste projeto. Irão ser estudadas soluções para dar resposta a fases do projeto detalhado como o sistema integrado de comunicações, sistema de navegação e sistema de monitorização e recolha de dados ambientais. Essas soluções serão facultadas através do estudo inicial dos sensores que podem equipar a plataforma, o projeto de um módulo de sensores essenciais à navegação do veículo e do planeamento e dimensionamento do sistema de telecomunicações.

Tendo em vista o sucesso do projeto detalhado, foram definidos os seguintes quatro principais objetivos: estudo e análise dos equipamentos que devem equipar os diferentes módulos de sensores do veículo; dimensionamento e otimização de um sistema de telecomunicações capaz de cumprir as especificações impostas; prova de conceito através da integração dos dois sistemas; realização de testes experimentais do funcionamento da plataforma.

De forma a possibilitar os estudos mencionados e cumprir com os objetivos supramencionados, foi necessário estruturar o trabalho em seis capítulos. Inicialmente será feito um enquadramento do projeto, que está vertido no presente capítulo. De seguida, no segundo capítulo, irá ser feito um estudo do tipo de sensores existentes no mercado, sendo feita uma descrição básica do seu princípio de funcionamento; irá incluir um estudo das diferentes interfaces existentes entre sensores; uma análise de veículos já existentes, de forma a perceber quais os sistemas que estes utilizam; finalmente será feita a escolha dos sensores que irão equipar a plataforma. No terceiro capítulo irá ser estudado

o sistema de telecomunicações, sendo feito o dimensionamento e testes das antenas. De forma a unir os estudos efetuados anteriormente, irá ser feita uma integração dos sistemas (capítulo 4), conseguindo assim fazer uma prova de conceito através dos testes à plataforma (capítulo 5). Para finalizar, irei apresentar uma análise de resultados e as respetivas conclusões.

2 Sensores de um Veículo Não Tripulado

2.1 Introdução

Um Veículo Não Tripulado de Superfície é definido principalmente pelo seu nível de automação, podendo este ser completamente autónomo ou simplesmente não tripulado, ou seja, com nível de automação mais baixo. Este nível de automação está diretamente relacionado com a sua capacidade de controlo, navegação e orientação (Liu et al., 2016). É expectável que nos próximos anos estes veículos produzam enormes benefícios, tais como o baixo custo de operação e desenvolvimento, o aumento de segurança a nível pessoal e a extensão do alcance operacional e precisão. Com o auxílio de equipamentos de navegação (unidade inercial e GPS) mais eficazes e acessíveis, e equipamentos de comunicações sem fios mais poderosos, estes veículos tornam-se capazes de cumprir qualquer tipo de missão (Manley, 2008).

Neste capítulo, irá ser feito o estudo dos tipos de sensores existentes no mercado, de forma a perceber quais os que devem equipar o veículo de superfície. Para o fazer, temos de ter em conta quais os pré-requisitos definidos inicialmente, pois o tipo de sensores irá variar consoante o tipo de missões que o veículo terá de desempenhar. Tendo em conta as missões e operações mencionadas no capítulo anterior, deduzem-se as seguintes tarefas (IOA – Conceito de Emprego Operacional de Veículos Não Tripulados de Superfície) (Estado Maior da Armada, 2016):

- 1) Caraterização ambiental (perfil de praia, topografia e oceanografia);
- 2) Recolha de informação, vigilância, reconhecimento tático e operacional;
- 3) Apoio ao comando, controlo e comunicações (C3) através de relé e transporte de dados;
- 4) Detecção de contactos (indivíduos e embarcações);
- 5) Seguimento automático de contactos (registo radar, laser e multiespectral);
- 6) Transporte e libertação de meios de salvamento;

- 7) Detecção de submarinos (registro multiespectral, acústico e anomalia magnética);
- 8) Detecção de minas (registro multiespectral, anomalia magnética);
- 9) Detecção de frequências eletromagnéticas;
- 10) Aplicação de medidas ativas (capacidade de *jamming*);
- 11) Avaliação do estado de infraestruturas (registro radar, laser e multiespectral);
- 12) Levantamento hidrográficos;
- 13) Levantamentos oceanográficos;
- 14) Detecção e seguimento de hidrocarbonetos;
- 15) Levantamento topográficos;
- 16) Observações meteorológicas;
- 17) Detecção radiológica, biológica e química.

Após análise das tarefas deduzidas para o veículo autónomo, foi idealizado um sistema por módulos, de forma a possibilitar o empenhamento em todas as operações planeadas. Em seguida é apresentada uma lista de todos os módulos idealizados, havendo um módulo essencial, o módulo de navegação.

- 1) Módulo essencial à navegação (GPS, IMU, RADAR, Sonda);
- 2) Módulo Visual (Câmara EO, Imagem térmica, LIDAR);
- 3) Módulo Subaquático (Sonares, sensores de poluição);
- 4) Módulo Meteorologia;
- 5) Módulo de Guerra Eletrónica;
- 6) Módulo NRBQ (Nuclear Radiológico Biológico e Químico).

Durante o capítulo, inicialmente, será feita uma análise teórica de todos os equipamentos que foram considerados necessários, à exceção dos módulos 5 e 6, que devido à sua particular complexidade extravasam o tema desta dissertação. De seguida, no segundo subcapítulo, irá ser abordado os diferentes tipos de interface que equipam os vários sensores, de forma a perceber como eles se interligam e como se podem interligar ao equipamento de controlo. O terceiro subcapítulo inclui uma análise do “estado da arte”, ou seja, um levantamento de Veículos de Superfície Não Tripulados já existentes e de quais os sensores que os equipam. Para terminar, será feito um estudo de mercado para

os sensores estudados no primeiro subcapítulo, permitindo assim a escolha dos equipamentos para os módulos de sensores do *USV*.

2.2 Tipos de Sensores

Nesta secção irá ser feita uma introdução teórica aos sensores que podem equipar o veículo, incidindo sobre o tema com particular foco no GPS e IMU, que serão usados para a implementação prática de um módulo de navegação básico, que irá ser descrito no capítulo 4.

2.2.1 Módulo de Navegação

O módulo de navegação é o sistema de sensores de maior importância para o Veículo de Superfície Não Tripulado pois é neste que vai assentar a sua capacidade de operação, ou seja, este módulo confere ao veículo a capacidade de se deslocar de forma autónoma, sabendo a sua localização, movimentações e panorama à sua volta, permitindo assim uma eventual automação total ou operação remota.

O módulo é constituído por quatro sensores vitais: GPS (*Global Positioning System*), que dá a posição absoluta; IMU (*Inertial Measurement Unit*), que fornece a orientação que, em conjunto com o GPS disponibiliza uma melhor estimativa da posição e velocidade; RADAR (*Radio Detection and Ranging*), serve para detetar obstáculos à superfície; e a sonda serve para medir a profundidade atual do fundo marítimo.

2.2.1.1 GPS

O GPS é um sistema de navegação que utiliza satélites desenvolvido pelo Ministério da Defesa dos Estados Unidos na década de 70, de forma a equipar as forças militares americanas com o sistema, cumprindo assim as suas necessidades. Anos mais tarde, esta tecnologia tornou-se também acessível ao mundo civil. Este sistema fornece continuamente a posição, em coordenadas, e informação temporal em qualquer lugar do planeta e em quaisquer condições meteorológicas (El-Rabbany, 2002).

Pode explicar-se o princípio de funcionamento do sistema recorrendo aos seus diferentes segmentos, percebendo como estes funcionam e como contribuem para o funcionamento em geral do sistema. Os segmentos são: segmento espacial, segmento de controlo e segmento utilizador. **O segmento espacial** é composto por uma constelação de 24 satélites posicionados a cerca de 20200 km acima da superfície terrestre, distribuídos por seis planos orbitais aproximadamente circulares, de modo a que em qualquer momento em qualquer local da superfície terrestre se encontrem no mínimo quatro satélites. As estações de rastreio observam os satélites e enviam essa informação para a estação de controlo principal. Na **estação de controlo** principal a informação é processada e enviada para os satélites sob a forma de mensagens de navegação. Os objetivos das estações passam pela verificação do funcionamento de cada satélite, calcular e prever as orbitas, controlar as manobras de substituição dos satélites, sincronizar os relógios, determinar parâmetros ionosféricos e atualizar as mensagens de navegação. A receção dos sinais enviados pelos satélites é conseguida por todos os recetores (**segmento do utilizador**) de sinais de satélites da constelação e seus operadores, em qualquer localização geográfica, à superfície terrestre. Um recetor GPS é constituído por: antena com pré-amplificador de sinal cuja função é captar os sinais emitidos pelos satélites; recetor propriamente dito, que integra os componentes de software e hardware indispensáveis ao funcionamento do sistema; e um relógio com oscilador de quartzo.

A necessidade de existência de um mínimo de quatro satélites visíveis prende-se com o facto de ser este o número mínimo de satélites cuja visibilidade é imprescindível para navegação e posicionamento. Cada um dos satélites funciona como um ponto de referência, a partir do qual os recetores de sinal, à superfície, definem a sua posição. Com base nas distâncias calculadas a 4 satélites é possível calcular a posição do recetor à superfície da terra com base num processo de trilateração (a trilateração envolve a medição de distâncias a um recetor GPS, através da contabilização do tempo de percurso de sinais de rádio que multiplicado pela velocidade das ondas eletromagnéticas contabiliza a distância, denominada pseudo-distância). A terminologia pseudo-distância, utilizada para referir a distância do recetor ao satélite, prende-se com o facto de a mesma ser o resultado da acumulação de vários erros.

Os erros de localização mencionados anteriormente podem ser erros nos relógios dos satélites e do recetor e erros causados pela passagem do sinal na atmosfera terrestre, que podem rondar as dezenas de metros, segundo El-Rabanny. Para contornar tal situação foram desenvolvidos sistemas para aumentar a precisão da localização através de correções enviadas por estações em terra. Estes sistemas estão divididos em três grupos: *DGPS* (*Differential Global Positioning System*), *SBAS* (*Satellite-based Augmentation Systems*) e *RTK* (*Real-Time Kinematics*). O modo de funcionamento dos três sistemas é assente no mesmo princípio: uma estação em terra que fornece correções aos erros gerados, sendo que diferem na forma como os calculam e na forma como distribuem as mensagens que contêm os erros.

O *DGPS*, como o nome indica, utiliza duas antenas recetoras: uma fixa com posição conhecida (*RS – Reference Station*) e outra móvel (*rover*) cuja localização é obtida por um GPS e mais tarde corrigida, ver figura 2.1. As estações de referência utilizam transmissores rádio na banda dos 283,5 – 325 kHz, normalmente instalados em faróis ou zonas costeiras (figura 2.2) para enviar mensagens no formato *RTCM*⁵ contendo informações para a correção das pseudo-distâncias obtidas pelo *GPS* móvel, assim como correções de relógio e efemérides dos satélites (El-Rabbany, 2002).

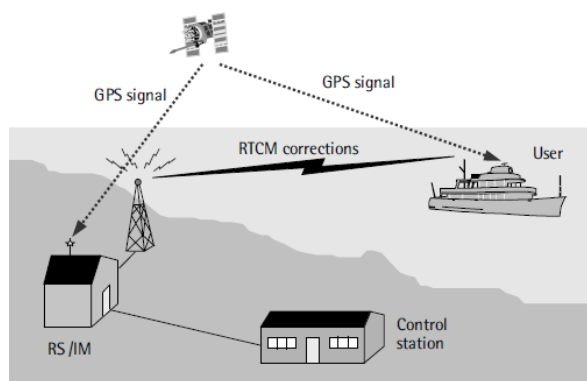


Figura 2.1: Funcionamento de uma estação *DGPS*⁶

⁵ *Radio Technical Commission for Maritime Services*

⁶ Retirado de (El-Rabbany, 2002)

Localização	Alcance Teórico	Frequência
Sagres 37° 00' N 008° 00' W	200 NM	305,5 KHz
Cabo Carvoeiro 39° 21' N 009° 24' W	200 NM	311,5 KHz
Horta (Faial) 38° 32' N 028° 37' W	200 NM	308,0 KHz
Porto Santo 33° 05' N 016° 20' W	200 NM	287,5 KHz

Figura 2.2: Constituição da rede *DGPS* Portuguesa⁷

SBAS é um sistema que, à semelhança do *DGPS*, utiliza correções fornecidas por estações em terra para melhorar a precisão da localização obtida por um *GPS*. As mensagens que contêm as correções são geradas da mesma forma, mas distribuídas por um processo alternativo. Este sistema recorre a várias estações de referência, que calculam as correções e as reencaminham para uma estação principal que combina todas as correções das diferentes estações, empacotando-as e enviando-as para um satélite geostacionário que por sua vez distribui para todos os utilizadores um conjunto de correções válidas para a área onde se encontra (figura 2.3) colmatando assim a desvantagem do sistema diferencial, pois a qualidade das correções não vai diminuído à medida que o utilizador se afasta da estação de referência. Na figura 2.4 pode observar-se os diferentes sistemas existentes para as diferentes áreas do planeta.

⁷ Retirado de <https://www.amn.pt/DF/Paginas/RedeDGPS.aspx>

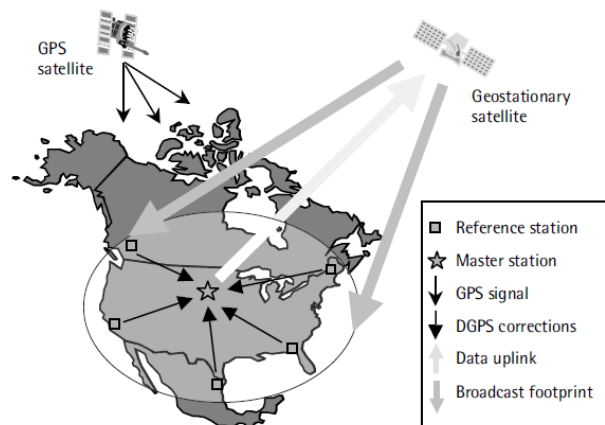


Figura 2.3: Funcionamento de um sistema *SBAS* (WAAS)⁸

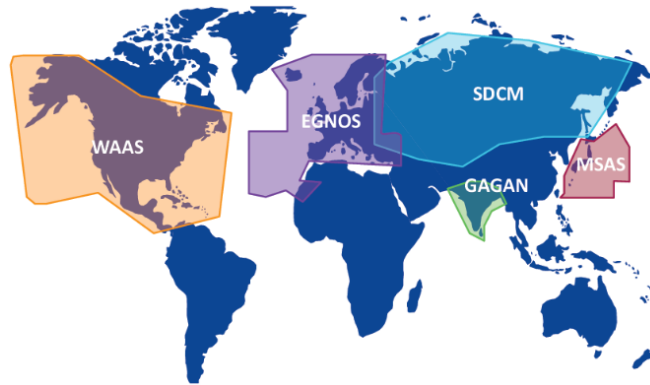


Figura 2.4: Diferentes sistemas *SBAS* consoante a região do utilizador⁹

RTK é o mais recente sistema de melhoria de precisão para um sistema GPS e, à semelhança do *DGPS*, utiliza radiocomunicações para enviar as correções à estação móvel a partir da estação de referência. Pode também utilizar a internet como meio de disseminação dessas correções. A principal diferença, relativamente aos sistemas anteriores, reside no método de cálculo das correções. O *RTK* utiliza informações da fase da onda transportadora para calcular as correções às pseudo-distâncias obtidas, como é mostrado na figura 2.5, obtendo uma precisão centimétrica.

⁸ Retirado de (El-Rabbany, 2002)

⁹ Retirado de <https://www.gsa.europa.eu/european-gnss/what-gnss/what-sbas>

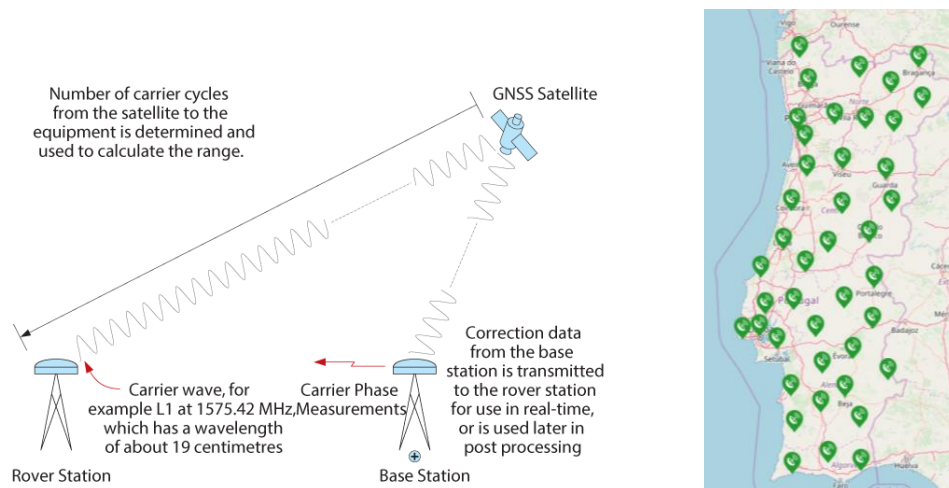


Figura 2.5: Funcionamento do sistema *RTK*¹⁰ e rede ReNEP¹¹ Portuguesa

2.2.1.2 IMU

O uso das IMU (*Inertial Measurement Unit*) surgiu em 1930, tendo sido utilizado principalmente em sistema de navegação de aeronaves. Este equipamento é utilizado principalmente para medir a aceleração (a velocidade e a posição podem ser depois obtidas por integração numérica, mas sujeitas a uma incerteza crescente no tempo), orientação e a força do campo eletromagnético.

Atualmente, graças aos sistemas *MEMS* (*micro-electromechanical system*) estes instrumentos apresentam uma dimensão reduzida, com um baixo custo de produção e baixo consumo de energia, o que permite uma grande área de aplicabilidade, quando comparados com as primeiras versões deste sistema, havendo ainda a opção da união do sensor *IMU* com o *GPS*, permitindo a otimização da localização em qualquer lugar no planeta (Ahmad, Ghazilla, Khairi, & Kasi, 2013).

¹⁰ Retirado de <https://www.novatel.com/an-introduction-to-gnss/chapter-5-resolving-errors/real-time-kinematic-rtk/>

¹¹ ReNEP é um serviço público de geoposicionamento prestado pela Direção Geral do Território para posicionamento em tempo real, utilizando a técnica RTK

Estes instrumentos apresentam-se divididos em dois, acelerómetro e giroscópio, ou três sensores, acelerómetro, giroscópio e magnetómetro, normalmente com 2 ou 3 graus de liberdade cada sensor. O acelerómetro mede a aceleração inercial, o giroscópio mede a rotação angular e o magnetómetro mede a direção das forças magnéticas, melhorando assim as leituras do giroscópio, pois compensa o grande desvio de leituras que ocorre no giroscópio devido às alterações de temperatura. A presença do magnetómetro só será benéfica se o ambiente onde são realizadas as medições não for rico em metais ferromagnéticos. Caso contrário, todas as leituras sofrerão erros (Ahmad et al., 2013).

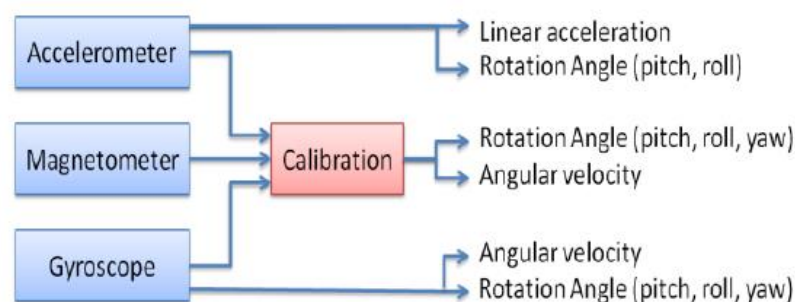


Figura 2.6: *IMU* composto por 3 sensores¹²

Aquando a escolha de um sensor *IMU* devemos considerar os seguintes fatores: **dimensão**, pois os produtos finais com vista a mobilidade, como por exemplo os telemóveis, têm de ser cada vez mais pequenos e leves; **Precisão de leituras**, os desvios nas leituras ao longo do tempo por parte dos giroscópios podem ser influenciados por efeitos de temperatura, *bias* (erro sistemático provocado pela aceleração da gravidade), quantização ou calibração, podendo estes ser diminuídos, juntamente com as leituras dos acelerómetros, através de técnicas de filtragem; **Taxa de refrescamento**, em que dependendo da aplicação, a taxa com que são atualizadas as leituras é muito importante pois permite ou não a atualização dos dados no período de tempo mais indicado; **Graus de liberdade**, que são determinados pelo número de parâmetros independentes dentro de cada sistema, ou seja, a quantidade de eixos que cada sensor consegue realizar medidas, sendo que pode variar entre 2 e 9 graus de liberdade nos *IMU*.

¹² Retirado de (Ahmad et al., 2013)

2.2.1.3 RADAR

Radio Detection and Ranging (RADAR) é um sistema que envolve a transmissão de uma onda eletromagnética até a um alvo de interesse, a sua reflexão no objeto e a recepção da onda refletida de volta no sistema. Após este processo, este sinal sofre um pós-processamento para retirar as informações desejadas (figura 2.7). Este sistema foi desenvolvido originalmente para fazer a detecção de aeronaves inimigas durante a 2ª Grande Guerra, tendo já sido estudado este fenómeno por parte de Heinrich Hertz e James Maxwell no século XIX. Hoje em dia e ao longo dos tempos, são diversas as aplicações de um radar: detecção de inimigos, controlo de tráfego aéreo, identificação de objetos, detecção da velocidade de veículos, mapear calotas polares, mapear desflorestação, meteorologia, entre outras (Melvin & Scheer, 2014).

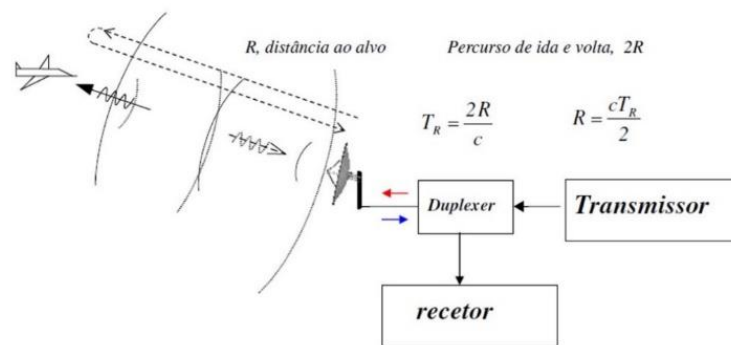


Figura 2.7: Distância a um alvo num Radar de Impulsos¹³

O tipo de aplicação ou a sua banda de funcionamento, segundo Skolnik, define a classificação do tipo de radar (figura 2.8): *Pulse Radar*, é um radar que emite impulsos retangulares; *Pulse Compression Radar*, usa impulsos longos e através de modulação interna obtém a energia de impulso longo com a resolução de um impulso curto; *Continuous Wave (CW) Radar* utiliza ondas sinusoidais contínuas detetando alvos em movimento através do efeito de Doppler; *MTI (Moving Target Indicator)*, é um radar de impulsos que utiliza uma baixa taxa de repetição de impulsos (PRF), detetando alvos em

¹³ Retirado de (Moreira, 2018)

ambientes com ruído¹⁴; *Pulse Doppler Radar*, utiliza as alterações na frequência (*Doppler*) para detetar alvos em ambiente com ruído, usando PRF elevados ou médios; *Synthetic Aperture Radar (SAR)* é um tipo de radar que utiliza a informação da fase do sinal refletido aliado à compressão de impulsos para obter uma imagem radar de alta qualidade, simulando a sua abertura com o movimento do próprio radar. Para além dos mencionados anteriormente, existem ainda mais tipos de radar que derivam dos supramencionados, como *FMCW*, *High resolution radar*, *ISAR*, *Guidance radar*, *Weather observation*, *Doppler weather radar*, *Target recognition*, entre outros (Skolnik, 1990).

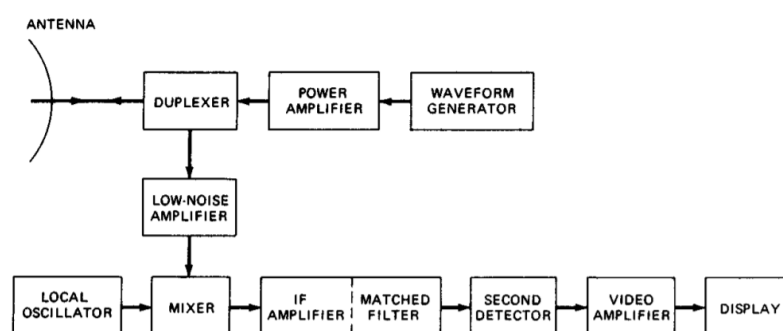


Figura 2.8: Diagrama de blocos de um sistema radar¹⁵

2.2.1.4 Sonda

A sonda é um equipamento utilizado em embarcações com o objetivo de fazer medições de profundidades da água, desde o local onde se encontra a sonda (normalmente a quilha) até ao fundo do mar. Este sensor utiliza ondas sonoras que viajam no meio aquático, de forma a fazer a medição do tempo de propagação da onda sonora, desde o sensor até aos objetos. O tempo total gasto desde a transmissão até à receção do eco equivale à distância percorrida, conhecida a velocidade do som na água (R.Sathishkumar, T.V.S Prasad Gupta, & M.Ajay Babu, 2013).

¹⁴ Zonas com interferências provenientes de chuva, maretas ou grandes concentrações de alvos mais pequenos (*Clutter*).

¹⁵ Retirado de (Skolnik, 1990)

O principal constituinte de uma sonda é o transdutor, elemento responsável pela transmissão e recepção das ondas sonoras. Um transdutor é um conversor de energia que oscila na sua frequência de ressonância quando a energia é aplicada no equipamento em análise, transmitindo essa energia para o ambiente que o rodeia, neste caso água, e criando assim ondas sonoras que são propagadas no meio. No momento de recepção, o processo é recíproco. Existem três tipos de transdutor: *electrostrictive*, *piezoelectric resonator*, e *magnetostrictive* (R.Sathishkumar et al., 2013).

2.2.2 Módulo Visual

O módulo visual é um dos elementos que irá complementar o módulo de navegação, pois vai não só confirmar as informações reunidas pelo módulo de navegação, mas também permitir a eventual operação à distância deste veículo e ainda aumentar a sua capacidade no que diz respeito ao desempenho de diferentes missões.

O módulo em questão pode ser constituído por três sistemas: uma câmara tradicional, ou seja, câmara de imagem no espectro do visível, uma câmara de imagem térmica e uma câmara LIDAR (*Light Detection and Ranging*).

2.2.2.1 Câmara EO/IR

Um sistema eletro-ótico é um sistema que contém sensores capazes de captar radiação eletromagnética e transformá-la numa imagem. O espectro eletromagnético está dividido em várias gamas de radiação. Ainda que o ser humano seja apenas sensível à radiação visível, existem outras gamas de radiação passíveis de ser usadas por ele com recurso a algumas ferramentas de ajuda. A radiação infravermelho e visível são as mais utilizadas em imagiologia e vigilância.

O funcionamento e composição de uma câmara tradicional (espectro visível) e de uma câmara térmica (espectro infravermelho) é relativamente simples e semelhante, diferindo principalmente num componente: o sensor de imagem.

A unidade de processamento em sistema de captura de imagem podem ser um *CPU* (*Central Processing Unit*) genérico, *DSP* (*Digital Signal Processor*), *FPGA* (*Field Programmable Gate Array*) ou processadores dedicados, não havendo nenhuma

especificidade acerca de que tipo de câmara deve equipar determinado tipo de unidade de processamento. A principal diferença reside no sensor de imagem, que capta a energia proveniente das ondas eletromagnéticas que chegam até ele. Para câmaras no espectro visual existem dois grandes tipos de sensores: *CMOS* (*Complementary Metal-Oxide Semiconductor*) e *CCD* (*Charge-Coupled Device*). A principal vantagem do sensor de imagem *CMOS* relativamente ao *CCD* é a sua capacidade de juntar num simples chip (pixel) o sensor e o processamento de imagem devido ao seu reduzido tamanho. A principal vantagem do sensor *CCD* é a produção de excelente qualidade de imagem, em troca de um maior custo (Belbachir, 2010).

Nas câmaras de imagem térmica (figura 2.10), o sensor responsável pela conversão de energia térmica em energia elétrica é, usualmente, um microbolómetro. Estes dispositivos são formados por uma matriz de vários elementos, cada elemento é formado por três camadas. A primeira camada é constituída por um elemento absorvente que recebe a radiação eletromagnética, variando a sua resistência elétrica consoante a quantidade de energia, fazendo variar, por sua vez, a resistência elétrica que possui, e permitindo desta forma saber a temperatura e a potência da radiação incidente. A segunda camada é constituída por espaço vazio, para que não exista contacto entre o elemento absorvente e o circuito integrado. A terceira camada é constituída por um circuito integrado mergulhado num substrato, responsável pelas leituras. Estes sensores são leves, baratos e consomem pouca energia, comparativamente a alternativas mais caras, que apresentam sistema de refrigeração, menos ruído e maior resolução.

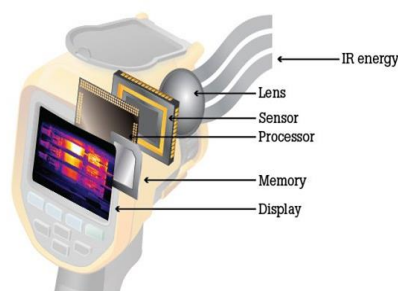


Figura 2.9: Constituição de uma câmara IR (espectro infravermelho)

2.2.2.2 Sistema LIDAR

LIDAR (Light Detection and Ranging) é um sistema, cuja invenção remonta aos anos 60, que partilha o fundamento de funcionamento com o RADAR, diferindo principalmente no tipo de radiação eletromagnética que utiliza para fazer a detecção de objetos. Este sistema utiliza um método que determina a distância a um objeto iluminando-o com um laser e medindo a sua reflexão com um sensor, sistema utilizado mais recentemente em carros autónomos para localizar objetos em redor do veículo, ver figura 2.11 (Dubayah & Drake, 2000).

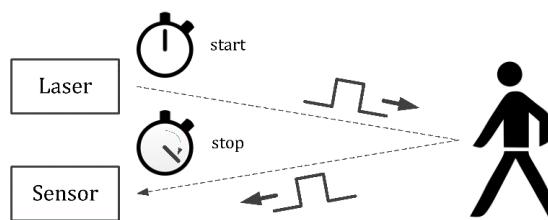


Figura 2.10: Determinação da distância ao objeto – *LIDAR*

Para além da medição de distâncias a objetos é possível fazer a representação digital em três dimensões de um objeto, assim como a criação de mapas de alta resolução através de diferenças em tempos de chegada, comprimento de onda e intensidade das ondas refletidas. Este processo está assente na fusão sensorial entre GPS, IMU e LIDAR, o que permite a criação de uma nuvem de milhões de pontos referenciados geograficamente, possibilitando assim a construção de mapas com múltiplas camadas como representado na figura 2.12 (Center, 2012).



Figura 2.11: Mapa de alta resolução

2.2.3 Módulo Subaquático

As operações para o projeto VENT-SUP-EN, mencionadas anteriormente, fazem prever vários cenários onde se necessita empenhamento de sensores capazes de fazer leituras em ambiente subaquático, o que implica o aprestamento do veículo com um módulo que contenha equipamentos capazes de o fazer.

De forma a cumprir com o acima referido, os sensores previstos são sonares e sensor de poluição, capaz de detetar e seguir hidrocarbonetos e outras substâncias presentes no meio aquático.

2.2.3.1 SONAR

SONAR (*Sound Navigation and Ranging*) é um instrumento cujo propósito é a deteção e medição da distância a objetos. Tal como os sistemas *RADAR* e *LIDAR*, utiliza o tempo de propagação de uma onda eletromagnética para fazer os cálculos mencionados (figura 2.13), sendo que cada um destes sistemas utiliza uma porção do espectro eletromagnético diferente. O SONAR, como o próprio nome indica, utiliza ondas sonoras que se propagam no meio aquático. Esta tecnologia surgiu em 1912, tendo-se massificado e aprofundado conhecimentos sobre a matéria durante a primeira guerra mundial, com o intuito de localizar submarinos (Carter, Mehta, & McTaggart, 2009). Existem dois principais tipos de sonar, o sonar ativo e o passivo, sendo que o primeiro emite ondas sonoras para

ativamente localizar objetos, e o passivo utiliza o som que os objetos emitem para determinar a sua localização.

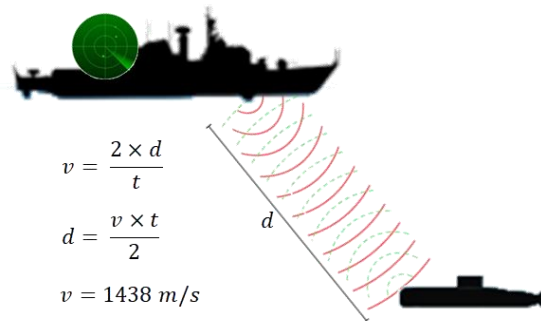


Figura 2.12: Funcionamento básico de um Sonar ativo

Atualmente, o sonar é utilizado para diversas atividades: navegação, pesca, estudos e pesquisa dos fundos oceânicos, guerra submarina, estudos de correntes marítimas, propriedades químicas e físicas da água, entre outras. Para cada área de utilização do sonar existem diferentes tipos de tecnologias: Sondas, que servem principalmente para medir a profundidade; Sonar Anti Colisão, que se caracteriza por servir, principalmente, para detetar obstáculos ou objetos de interesse no setor da proa da embarcação; Sonar Multifeixe, utilizado para fazer levantamentos hidrográficos ou batimetria; Sonar Lateral, que é semelhante ao multifeixe no que toca à sua função, mas que graças às suas características permite uma imagem de maior resolução, podemos ver essa comparação na figura 2.14; *ADCP (Acoustic Doppler Current Profiler)* é um equipamento que utiliza o efeito de Doppler para fazer medições na coluna de água; *Sub bottom profiler sonar*, é um sonar que utiliza frequências mais baixas (para permitir penetração no solo) para fazer estudos das camadas sedimentares do fundo oceânico.

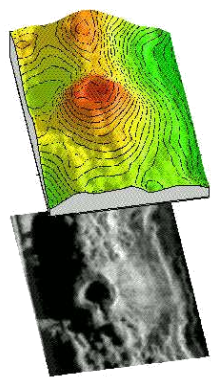


Figura 2.13: Imagem obtida por um sonar de multifeixe (cima) vs sonar lateral (baixo)

2.2.3.2 Sensor de Poluição

Uma das operações com enorme relevância a ser desempenhada é a detecção de poluição. Existem algumas medições, através de processos químicos, que podem ser levadas a cabo para detetar indícios de poluição nas águas portuguesas. A medição do PH, níveis de amoníaco, quantidade de nitrato, quantidade de cloretos, turvação, condutividade e temperatura são alguns dos processos que fazem parte do despiste de poluição. O seguimento de hidrocarbonetos é uma das tarefas com maior impacto para a manutenção das águas. O sensor de poluição utiliza um fotodíodo UV de interferência e, através do método de fluorescência, mede o nível de óleos no ambiente em que está inserido.

2.2.4 Observações Meteorológicas

Uma das formas de reduzir o risco no empenhamento de pessoal é o antecipado conhecimento das condições meteorológicas.

Os diferentes sensores que podem compor esta estação são sensores de temperatura, de humidade, pressão e velocidade e direção do vento. O funcionamento deste tipo de sensores depende essencialmente do modelo adquirido, pois existem várias técnicas que podem ser implementadas para efetuar leituras.

2.3 Interfaces de comunicação

Entre qualquer tipo de sensor, ou sensor e unidade de controlo, existem meios de comunicação específicos que fazem parte das suas características, possibilitando assim a capacidade de transferência dados entre si, de forma a criar uma rede de informação.

Neste subcapítulo irá ser feita uma breve descrição dos principais tipos de interfaces existentes, explicando o princípio de funcionamento e principais características que os distinguem, o que ajudará mais tarde na escolha dos sensores, para que seja construída uma rede de sensores robusta e o mais eficiente possível.

2.3.1 RS-232

RS-232 é um tipo de ligação ponto a ponto de série de transferência de dados, permitindo uma comunicação máxima a poucas centenas de metros. Utiliza transmissão diferencial, com valores típicos -3 a -25 V e +3 a 25V, com uma velocidade máxima de 115 kbps. É habitual utilizar conectores DB-9, sendo por vezes usados conectores DB25 (Damas, 2019).

2.3.2 RS-422

A ligação RS-422 é um tipo de comunicação usada normalmente para estender o alcance das comunicações RS-232, pois é possível atingir uma distância de 1500m. Utiliza transmissão *full-duplex* diferencial, -6V e +6V (valores máximos admitidos), com par entrelaçado, podendo atingir os 10Mbit/s. A ligação pode ser feita de ponto a ponto ou *multi-drop* (1 transmissor com até 10 recetores) (Damas, 2019).

2.3.3 RS-485

A ligação RS-485 é um tipo de comunicação usada normalmente em redes de campo, sendo tipicamente possível atingir uma distância de 1200m. Utiliza transmissão *half-duplex* diferencial, -7V e +12V (valores máximos admitidos), com par entrelaçado, podendo atingir os 35Mbit/s. A ligação pode ser multiponto (até 32 emissores/recetores) (Damas, 2019).

2.3.4 Serial Peripheral Interface

A comunicação SPI usa uma arquitetura *master-slave*, sendo esta *full-duplex* e síncrona, sem velocidade definida. Este tipo de ligação é usada normalmente em sistemas embebidos, estando estes a uma curta distância (memória flash, sensores, conversores A/D, entre outros). Para estabelecer este tipo de comunicação são usados 4 fios: *Serial Clock* (SCLK), *Master Output Slave Input* (MOSI), *Master Input Slave Output* (MISO) e *Slave Select* (SS) (Damas, 2019).

2.3.5 Inter-Integrated Circuit

A comunicação I^2C usa uma arquitetura *master-slave* (capacidade para vários), sendo esta *half-duplex* e síncrona, com velocidades pré-definidas (100 Kb/s; 400 Kb/s e 3,4 Mb/s). Este tipo de ligação é usada normalmente em sistemas embebidos, estando estes a uma curta distância. Para estabelecer este tipo de comunicação são usados apenas 2 fios: *Serial Clock Line* (SCL) e *Serial Data Line* (SDA). Sistema de endereçamento de 7 ou 10 bits, com conjuntos de dados de 8 bits antecipados do endereço (Damas, 2019).

2.3.6 NMEA

National Marine Electronics Association (NMEA) é a associação que dá nome ao protocolo de comunicações de equipamentos de navegação. Existem duas versões deste protocolo, NMEA 0183 e NMEA 2000, mais recente. A versão de 2000 é bidirecional com uma rede com múltiplos transmissores e múltiplos recetores, sem um controlador central, diferenciando-se assim da versão 0183 (Parthasarathy, 2006).

Este protocolo é assíncrono, com elementos que “falam” ou “ouvem”, tendo por base o protocolo RS-422 para o NMEA 0183 e o protocolo CAN para o NMEA 2000. Os equipamentos comunicam entre si nesta rede NMEA, através de mensagens com elementos identificativos de início, fim e caracterizador de tipo e origem da mensagem.

2.4 Trabalhos Relacionados

Tendo em conta as variadas operações em que os Veículos Não Tripulados têm de participar, os níveis de autonomia, automação e capacidade de multitarefa são elementos essenciais no que concerne o desenvolvimento de um VENT-Sup, sendo assim possível observar estas características em USVs existentes.

ROAZ II é o nome do veículo de superfície autónomo desenvolvido pela LSA (Laboratório de Sistemas Autónomos) que pertence ao ISEP (Instituto Superior de Engenharia do Porto). Este veículo é um catamarã com propulsão elétrica, desenvolvido para operar em mar aberto e desempenhar vários tipos de missões, graças à estrutura entre cascos. As principais missões para que foi desenhado são operações oceanográficas, recolha de dados ambientais, operações de busca e salvamento e de defesa portuária (Almeida et al., 2009).

A recolha de dados ambientais, mais especificamente, a análise da qualidade da água é uma das principais missões do **HYCAT**. Um USV em forma de catamarã, de reduzidas dimensões, próprio para operar em águas calmas que permite, através dos seus sensores e sistema de telecomunicações, fazer análise, levantamento e transmissão de dados para o controlador que se encontra em terra (YSI, 2019b).

Uma das principais missões da Marinha Portuguesa é a salvaguarda da vida no mar, por isso é necessário desenvolver e aplicar o máximo conhecimento no que concerne as operações de busca e salvamento marítimo. **UCAP** é uma cápsula que transporta uma jangada de salva-vidas até ao local de interesse por um USV de maiores dimensões. A cápsula possui meios de localização e guiamento que permitem o emprego da balsa no local desejado (Younes, 2015).






Um USV pode ser usado isoladamente para desempenhar a missão desejada ou pode também atuar como um multiplicador de força, ou seja, ser usado em conjunto com um navio comum para desempenhar dada missão.

Uma das principais missões, com enorme relevância para qualquer marinha, são os levantamentos hidrográficos. Os veículos não tripulados, dependendo da sua

configuração, participam ativamente no cumprimento destas missões. **C-Worker 5** é uma plataforma concebida pela ASV GLOBAL que participou em conjunto com um navio num levantamento hidrográfico de 9630km, tendo a plataforma contribuído com 44% do total do projeto e poupado desta forma cerca de 25 dias de operação (ASV, 2017).

É de salientar que estes veículos conseguem desempenhar uma grande variedade de diferentes missões, sendo apenas necessário configurar o *payload*. **CAT-Surveyor** é um catamarã que possui um vasto *payload* que lhe permite fazer recolha de dados ambientais, levantamentos hidrográficos, defesa portuária, vigilância e monitorização (Subsea Tech, 2015).

Na tabela seguinte estão compiladas algumas das características (dimensões, sensores e comunicações) dos veículos de superfície mencionados anteriormente.

		ROAZ II	HYCAT	UCAP	C-Worker 5	CAT-Surveyor
						
Dimensões	Comp.	4,5 m	1,8 m	0,68 m	5,5 m	3 m
	Largura	2,2 m	0,86 m	0,38 m	1,7 m	1,6 m
	Altura	0,5 m	0,15 m	0,25 m	1,8 m	1,2 m
	Peso	200 Kg	53 Kg	30 Kg	1350 Kg	270 Kg
Comunicações		2.4 GHz (antena 12 dBi)	5,8 GHz	2.4 GHz (antena 7 dBi)	Rádio IP	2.4 GHz
Sensores	GPS	Septentrio PolaRx2	AtlasLink RTK GPS	NEO 6T GPS	(a)	(a)
	IMU	Microstain 3DM	(a)	PNI Trax AHRS	(a)	(a)
	RADAR	Lowrance 3G	(a)	(a)	(a)	(a)
	Câmaras	ThermalEye 5000XP; Bashler 1300ac;	(a)	(a)	(a)	Norbit iLIDAR
	SONAR	Imaginex DeltaT	YellowFin side scan sonar; YSI EXO2; SonTek M9	(a)	Reson 7101 multi-beam; Edgetech 4200 side-scan; Applanix PSMVs	Starfish 450; Teledyne BlueView; Norbit WBMS; ADCP Sontek

(a) Informações não disponibilizadas pelo fabricante






Tabela 2.1: Estado da Arte, VENT-SUP existentes

2.5 Escolha dos Sensores

No presente subcapítulo serão apresentadas opções de modelos de equipamentos existentes no mercado para cada um dos sensores que deverão equipar o veículo.

2.5.1 GPS

Na tabela 2.2 pode observar-se uma breve comparação das principais características técnicas de diferentes sistemas de posicionamento. Para fazer esta comparação foram escolhidos dois *GPS* que são utilizados, normalmente, em projetos de robótica e os restantes três utilizados em embarcações.

	U-blox NEO-M8N	GPS-RTK-NEO-M8P	FURUNO GP 320B	Lowrance Point-1	Garmin GPS 19x
Alimentação	3,6 V	3,3 V	12 – 24 V	9 – 18 V	9 – 16 V
Corrente de operação	100 mA	35 mA	55 – 105 mA	100 mA	40 mA
Precisão	2,5 m (2 m com SBAS)	2,5 m (0,025 m com RTK)	3 m	3 m	3 m
Taxa de refresco	5 Hz	0,25 – 10 Hz	1 Hz	10 Hz	10 Hz
Número de canais	50	72	12	32	32
TFF (<i>Time to First Fix</i>)	27 s	29 s	90 s	50 s	45 s
	1 s	1 s	20 s	3 s	1s
Aquisição de Sinal	1 s	1 s	(a)	5 s	2 s
Sistema de Correção de posição	WAAS, EGNOS e MSAS	RTK	WAAS	WAAS, EGNOS e MSAS	(a)
Interfaces de comunicação	SPI e UART	SPI e I ² C	NMEA 2000 e NMEA 0183	NMEA 2000	NMEA 0183
					
Preço	10,62 €	176€	379€	175€	239€

(a) Informações não disponibilizadas pelo fabricante

Tabela 2.2: Tabela Comparativa GPS¹⁶

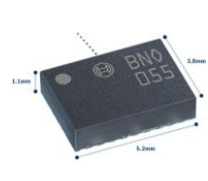



¹⁶ *Datasheets* referentes aos equipamentos: (Furuno, 2011; Garmin, 2012; Lowrance, 2018; u-blox, 2016)

Os sistemas acima apresentados apresentam diferentes valências, sendo todos eles ótimos sistemas a serem implementados no projeto VENT-SUP-EN. Fazendo uma análise pormenorizada e atendendo às características do projeto, é de realçar que os sistemas baseados no protocolo NMEA são dedicados a embarcações e prontos a ser ligados a uma rede NMEA de bordo. Contudo, apresentam um custo significativamente mais elevado. Ao analisar de forma pormenorizada os dois restantes sensores, podemos concluir que a principal diferença entre eles reside no sistema integrado de correção de posição, daí a disparidade de preço. A escolha do sensor será feita consoante o tipo utilização, ou seja, a tarefa que queremos desempenhar com o nosso projeto; se o nosso veículo tiver de fazer medições em espaço, como por exemplo, levantamentos hidrográficos numa lagoa, a melhor opção será um *GPS RTK*; se o nosso objetivo é simplesmente obter a posição do veículo ao longo do tempo numa viagem, a opção *GPS* com *SBAS* é perfeitamente viável.

2.5.2 IMU

Na tabela 2.3, são apresentados 4 sensores *IMU* e comparadas as principais características. Foram selecionados sistemas com 9 graus de liberdade, permitindo assim uma grande variedade de valores que podemos retirar destes sensores, tendo também sido escolhidos sistemas que possuem unicamente os 3 tipos de sensores: acelerómetro, giroscópio e magnetómetro, resultando assim num conjunto de sensores com uma grande capacidade de resposta a qualquer tipo de exigência que possa surgir no projeto em questão.

		BOSCH BNO055	InvenSense MPU 9250	STMicroelectronics LSM9DS1	SparkFun 9 DOF AHRS Module
Alimentação		2,4 – 3,6 V	2,4 – 3,6 V	2,4 – 3,6 V	3 – 5V
Corrente de operação	Acc.	130 μ A	450 μ A	600 μ A	145 μ A
	Giro.	5 mA	3,2 mA	4 mA	0,1 mA
	Magn.	500 μ A	280 μ A	600 μ A	100 μ A
Alcance	Acc.	± 2 g, ± 4 g, ± 8 g, ± 16 g	± 2 g, ± 4 g, ± 8 g, ± 16 g	± 2 g, ± 4 g, ± 8 g, ± 16 g	± 2 g, ± 4 g, ± 8 g, ± 16 g
	Giro.	$\pm 125^\circ$ /s, 250° /s, $\pm 500^\circ$ /s, $\pm 1000^\circ$ /s, $\pm 2000^\circ$ /s	$\pm 250^\circ$ /s, $\pm 500^\circ$ /s, $\pm 1000^\circ$ /s, $\pm 2000^\circ$ /s	$\pm 245^\circ$ /s, $\pm 500^\circ$ /s, $\pm 2000^\circ$ /s	$\pm 2000^\circ$ /s
	Magn.	± 1300 T (x,y), ± 2500 μ T (z)	± 4800 μ T	± 400 μ T, ± 800 μ T, ± 1200 μ T, ± 1600 μ T	± 800 μ T
Interfaces de comunicação		SPI, I ² C e UART	SPI e I ² C	SPI e I ² C	I ² C

Sensibilidade	Acc.	±2 g: 4096 LSB/g ±4 g: 2048 LSB/g ±8 g: 1024 LSB/g ±16 g: 512 LSB/g	±2 g: 16384 LSB/g ±4 g: 8192 LSB/g ±8 g: 4096 LSB/g ±16 g: 2048 LSB/g	±2 g: 16393 LSB/g ±4 g: 8197 LSB/g ±8 g: 4098 LSB/g; ±16 g: 1366 LSB/g	±2 g: 286 LSB/g ±4 g: 143 LSB/g ±8 g: 71 LSB/g ±16 g: 36 LSB/g
	Giro.	±125°/s: 262,4 LSB/°/s; ±250°/s: 131,2 LSB/°/s; ±500°/s: 65,5 LSB/°/s; ±1000°/s: 32,8 LSB/°/s; ±2000°/s: 16,4 LSB/°/s	±250°/s: 131 LSB/°/s; ±500°/s: 65,5 LSB/°/s; ±1000°/s: 32,8 LSB/°/s; ±2000°/s: 16,4 LSB/°/s	±245°/s: 114 LSB/°/s; ±500°/s: 57,1 LSB/°/s; ±2000°/s: 14,3 LSB/°/s	±2000°/s: 14,375 LSB/°/s
	Magn.	(a)	0,6 µT/LSB	±400 µT: 0,14 µT/LSB; ±800 µT: 0,29 µT/LSB; ±1200 µT: 0,43 µT/LSB; ±1600 µT: 0,58 µT/LSB	±800 µT: 0,73 µT/LSB
Variação da sensibilidade devido à temperatura	Acc.	±0,03 %/K	±0,026 %/°C	(a)	±0,01 %/°C
	Giro.	±0,03 - ±0,07 %/K	±4 %	(a)	±10 %
	Magn.	±0,01 - ±0,03 %/K	(a)	(a)	±0,3 %/°C
Taxa de Refresco		100 Hz	200 Hz	1000 Hz	(a)
					
Preço		9,94€	13,21€	5,36€	37,13€

(a) Informações não disponibilizadas pelo fabricante

Tabela 2.3: Tabela Comparativa IMU¹⁷





Através da análise da tabela anterior, podemos concluir que as diferentes unidades não apresentam grandes diferenças em termos de características e desempenho. Fazendo uma análise estritamente numérica, concluímos que o modelo da *STMicroelectronics* apresenta o melhor rácio entre desempenho e preço, com boas taxas de refresco e sensibilidade, não sabendo, ainda assim, como é que este se comporta com variações na

¹⁷ *Datasheets* referentes aos equipamentos: (Bosch, 2014; InvenSense, 2016; STMicroelectronics, 2015)

temperatura. Apesar do referido modelo ser uma boa opção pelas razões enunciadas, a escolha final recai sobre o modelo da *BOSCH*, uma vez que possui características muito semelhantes ao outro modelo, às quais acresce ainda uma corrente de operação mais baixa e uma maior variedade de interfaces de comunicação, que resulta numa maior versatilidade.

2.5.3 RADAR

Na tabela seguinte são apresentados radares de impulsos que podem equipar embarcações, de forma a fazer a deteção de alvos nas áreas envolventes. Com o intuito de equipar o projeto VENT-SUP-EN, foi feita uma breve comparação entre alguns sistemas radar colocando em perspetiva algumas das características que os diferenciam.

	Garmin Radome GMR™ 18 xHD	Raymarine Quantum Q24C	Furuno DRS4D-NXT	Lowrance 3G
Alimentação	11 – 35 V	12 – 24 V	12 – 24 V	9 – 31 V
Potência de transmissão	4 kW	20 W	25 W	(a)
Frequência de trabalho	9410 MHz	9354 – 9440 MHz	9380 – 9440 MHz	9300 – 9400 MHz
Rotações por minuto (RPM)	24 / 48	24	24 / 36 / 48	24 / 36
Largura de feixe Horizontal	5,2°	4,9°	2° - 3,9°	5,2°
Largura de feixe Vertical	25°	20°	25°	25°
Alcance máximo	48 mi	24 mi	26 mi	24 mi
Alcance mínimo	20 m	5,5 m	20 m	50 m
Descriminação em distância	16 m	6 m	20 m	50 m
Interfaces de Comunicação	NMEA 2000	NMEA 2000	NMEA 2000	NMEA 2000
				
Preço	1578€	1728€	1846€	1144€

(a) Informações não disponibilizadas pelo fabricante

Tabela 2.4: Tabela Comparativa RADAR¹⁸

¹⁸ *Datasheets* referentes aos equipamentos: (Furuno, 2016; Garmin, 2014; Lowrance, 2011; Raymarine, 2016)

Tendo em conta alguns dos conhecimentos teóricos apresentados anteriormente e as necessidades do projeto, de entre as opções apresentadas na tabela, o modelo da *Raymarine* é um dos equipamentos com melhor desempenho pelo preço, já que apresenta uma melhor discriminação em distância e menor distância de deteção, fatores de enorme relevância para este projeto. Para distância mais próximas da embarcação (aproximadamente 200 metros) e para cobrir a área que o radar convencional não é capaz de cobrir, é possível utilizar um sensor radar, usados também em veículos como prevenção de colisões, que utilizam a tecnologia *FMCW* para fazer a deteção de alvos a curtas distâncias.

2.5.4 Câmara EO/IR

Com o intuito de equipar o projeto VENT-SUP-EN foi feita uma breve comparação, tabela 2.5, entre algumas câmaras, colocando em perspetiva algumas das características que as diferenciam. O módulo visual é um dos elementos que irá complementar o módulo de navegação e irá permitir a eventual operação à distância deste veículo, aumentando ainda a sua capacidade no que diz respeito a missões de busca e salvamento, vigilância, patrulha de portos e zonas costeiras, análise de infraestruturas, entre outras missões de igual importância.

		HIKVISION DS-2A 720p IR PTZ	FLIR M132 Thermal Camera	FLIR DUO PRO R	AXIS Q8741 Bispectral	FLIR M400 Thermal Camera
Visível	Sensor	CMOS 1/3"	-	HD 1080p	CMOS 1/2,8"	HD 1080p
	Resolução	1280×720	-	1920×1080	1920×1080	1920×1080
	Zoom	23×	-	30×	30×	30×
	Focagem	4 - 92 [mm]	-	4,3 - 129 [mm]	4,3 - 129 [mm]	4,3 - 129 [mm]
IR	Sensor	-	Microbolometer	Microbolometer	Microbolometer	Microbolometer
	Resolução	-	320×240	336×256	384×280	640×480
	Zoom	-	2×	4×	(a)	4×
	Focagem	-	(a)	(a)	35 [mm]	35 - 105 [mm]
Interfaces de Comunicação		BNC	IP	HDMI; USB; PWM	SFP; RJ-45	IP; NMEA; RS-485; BNC
Consumo		20 W	15 W	(a)	16 W	50 W
Saída de Vídeo		NTSC; PAL	H.264	H.264	H.264	NTSC; PAL; H.264
Distância de deteção (Pequena embarcação)		(a)	1,3 km	(a)	3,2 km	6,9 km
		IP66	IP67	Sobreposição de imagens;	Contador de Pixéis (Deteção); IP66	Estabilizada; Lanterna; Integração com

			Radiometria; integração com GPS e IMU		Radar e AIS; IP66; Radiometria
					
Preço	659€	2233€	4020€	8935€	53615€

(a) Informações não disponibilizadas pelo fabricante

Tabela 2.5: Tabela Comparativa Câmara EO/IR¹⁹

Tendo em conta alguns dos conhecimentos teóricos apresentados anteriormente e as necessidades do projeto, das opções apresentadas na tabela, o modelo que apresenta melhor rácio qualidade / preço é: *FLIR DUO PRO R*, pois apresenta um sensor ótico com características semelhantes à melhor câmara apresentada na tabela; apesar do sensor térmico não ser o melhor entre as opções apresentadas, a capacidade de sobreposição de imagens faz com que isso seja compensado, pois temos o dobro do poder visual; sendo esta ainda capaz de ser integrada com *GPS*, o que permite localização geográfica em todas as imagens e ainda capaz de ser integrada com *IMU*, o que permite a estabilização da imagem. Os pontos negativos deste sistema são a falta de proteção ambiental, principalmente contra salpicos de água, e falta de um estabilizador giroscópico.

Tendo em conta que os aspetos negativos impedem que o sistema opere em ambiente marítimo, pois não tem proteção contra salpicos de água, a alternativa seria adquirir o equipamento da *AXIS*, pois apresenta sensivelmente as mesmas características técnicas acrescentando a proteção IP66, tendo como ponto negativo o preço mais elevado.

2.5.5 Sistema *LIDAR*

Com o objetivo de equipar o projeto VENT-SUP-EN foi feito uma breve comparação, tabela 2.6, entre alguns sistemas *LIDAR* colocando em perspetiva algumas das características que as diferenciam. Este sistema de ajuda à navegação é um dos elementos

¹⁹ *Datasheets* referentes aos equipamentos: (Axis, 2017; Flir, 2017b, 2017a, 2018; Hikvision, 2017)

mais importantes no que concerne a navegação, permitindo a eventual operação em modo autónomo complementando as informações do radar e câmaras relativas ao ambiente envolvente, fazendo a deteção de obstáculos próximos. Aumenta também a sua capacidade no que diz respeito a missões de busca e salvamento, vigilância, patrulha de portos e zonas costeiras, análise e mapeamento de infraestruturas, entre outras missões de igual importância.

	RPLIDAR A1M8	LIDAR-Lite v3HP	Hokuyo UST-10LX	RoboSense RS-LIDAR-16	Livox Mid-40
Alimentação	5 V	5 V	10 – 30 V	9 – 32 V	10 – 30 V
Alcance	0,2 - 12 m	0,5 – 40 m	0,06 – 10 m	0,2 – 150 m	0,2 – 260 m
Campo de visão (Horizontal)	0°-360°	20°	0° - 270°	0° - 360°	40°
Campo de visão (Vertical)	(a)	2°	2°	30°	38°
Resolução em distância	20 mm	35 mm	40 mm	20 mm	20 mm
Resolução angular	1°	(a)	0,25°	2°	0,1°
Frequência de amostragem	8000 Hz	> 1000 Hz	40 Hz	320 kHz	100 kHz
Frequência de digitalização	2 -10 Hz	(a)	4 Hz	5 – 20 Hz	(a)
Comprimento de onda do Lser	785 nm	905 nm	905 nm	905 nm	905 nm
Interfaces de Comunicação	UART	I ² C	RJ45	(a)	RJ45
Índice de Proteção	(a)	IPX7	IP65	IP67	IP55
					
Preço	89€	134€	1430€	3315€	535€

(a) Informações não disponibilizadas pelo fabricante

Tabela 2.6: Tabela Comparativa Sistema LIDAR²⁰

Tendo em conta alguns dos conhecimentos teóricos apresentados anteriormente e as necessidades do projeto, os modelos que melhor cumprem os requisitos são o modelo da *RoboSense* e o modelo da *Livox*, pois apresentam grande alcance e resolução, permitindo detetar objetos onde o Radar não consegue. Ambos permitem completar também o

²⁰ Datasheets referentes aos equipamentos: (Garmin, sem data-a; Hokuyo, 2018; Livox, 2019; RoboSense, 2019; RPLIDAR, 2016)

mapeamento de estruturas. A vantagem do modelo do *RoboSense* é o seu campo de visão de 360°, que possibilita uma total percepção do ambiente a toda a volta, enquanto que o modelo da *Livox* permite unicamente um ângulo de 20° para cada bordo a partir da linha de proa da embarcação, deixando uma grande zona de sombra. Analisando e tendo em conta o rácio desempenho / preço, o modelo *Mid-40* da *Livox* é o mais indicado, podendo melhorar o aspeto dos ângulos de visão. Ao optar pela gama acima da *Livox* (*Mid-100*), o campo de visão aumenta para os 100°, ficando pelo valor de 1335€.

2.5.6 Sistema Subaquático

Na tabela 2.7 são apresentados modelos exemplificativos, com características e preços indicativos para cada tipo de sensor. Tendo em conta a variedade de missões que o VENT-SUP-EN pode desempenhar, qualquer modelo apresentado poderá equipar o veículo dando assim a possibilidade de realizar todos os tipos de missão.

	Tipo	Alcance	Sensores	Preço
YSI EXO2	Sonda	250 m	<i>Barometer; Total Algae; Conductivity / Temperature; Depth; Dissolved Oxygen; pH; Turbidity</i>	247€

	Tipo	Alcance	Frequência	Campo de visão (Vertical)	Campo de visão (Horizontal)	Preço
Garmin Panoptix	Sonar anti colisão	100 m	417 kHz	0° - 90°	10°, 20°, 40° e 60°	1335€

	Tipo	Alcance	Resolução	Feixes	Frequência	Taxa de amostragem	Preço
Olex Atec 200	Sonar multifeixe	0,5 – 500 m	10 mm	256	100 – 800 kHz	50 Hz	53000€

	Tipo	Alcance	Resolução	Profundidade máxima de operação	Largura de Feixe	Preço
YellowFin	Sonar lateral	1000 m	Alcance / 1000	300 m	260 kHz: 2,2° × 75° 330 kHz: 1,8° × 60° 800 kHz: 0,7° × 30°	8020€

	Tipo	Alcance	Frequência	Potência de saída	Preço
Hondex HE-7300 DI	ADCP	5 – 2000 m	28, 32, 36, 40, 50, 65, 75, 100 e 200 kHz	600, 1200, 3000, 2500 W (depende da frequência)	2055€

	Tipo	Profundidade de	Profundidade de máxima	Frequência	Largura de feixe	Resolução da camada inferior	Preço
--	------	-----------------	------------------------	------------	------------------	------------------------------	-------

		penetração	de operação				
JW Fishers SBP-1	<i>Sub bottom profiler</i>	>30 m	150 m	10 kHz	30°	6 cm	16925€

Tabela 2.7: Tabela Comparativa Sistemas Subaquáticos²¹

2.6 Conclusão

Durante o presente capítulo foram abordados diferentes aspetos relativos ao sistema do VENT-SUP-EN. O estudo deste sistema iniciou-se com uma análise das necessidades sensoriais para o projeto, recorrendo aos pré-requisitos estabelecidos numa publicação referente a veículos não tripulados.

Tendo em vista as necessidades do projeto, apresentaram-se inicialmente os princípios de funcionamento da maioria dos sensores que podem vir a equipar o VENT-SUP-EN, incidindo em cada módulo e o que estes devem conter, tendo também sido abordadas as características dos principais protocolos e interfaces de comunicação dos sensores. Seguidamente, foi apresentado um quadro com informações sobre *USV* existentes no mercado com características idênticas ao projeto. Terminou-se o capítulo com a apresentação de estudos comparativos entre modelos de sensores.

O estudo efetuado ao longo do capítulo permitiu fazer uma análise inicial dos diversos sensores que poderão vir a equipar o veículo. Os temas abordados permitiram uma análise que abrangeu 12 das 17 tarefas previstas na introdução deste capítulo, ficando em falta: Apoio ao comando, controlo e comunicações (C3) através de relé e transporte de dados, que será abordado no próximo capítulo; transporte e libertação de meios de salvamento; deteção de frequências eletromagnéticas e capacidade de *jamming*; e deteção radiológica, biológica e química; que são tarefas que extravasam os objetivos desta dissertação.

²¹ *Datasheets* referentes aos equipamentos: (Garmin, sem data-b; Hondex, 2012; Imagenex, 2004; JW Fishers, 2010; Olex, 2018; YSI, 2019a)

3 Sistema de Telecomunicações

3.1 Introdução

Telecomunicação, segundo o *ITU-R*²², é qualquer transmissão, emissão ou recepção de sinais, escrita, imagens ou informação de qualquer género por fio, radio, ótico ou outro sistema eletromagnético (International Telecommunication Union – Radiocommunication Sector, 1998). No caso em estudo, interessam as telecomunicações por radiofrequência, um sistema que utiliza radiação não ionizante do espectro eletromagnético, de frequências entre os 3 kHz e os 300 GHz (Hitchcock, 2012).

Um sistema de radiocomunicações, representado esquematicamente na figura 3.1, é constituído por um emissor, um recetor, dois guias de ondas, duas antenas e um canal de propagação.

O canal de propagação nas radiocomunicações é a atmosfera, que possui características que permitem a propagação das ondas eletromagnéticas. No entanto, existem fatores que degradam a qualidade do sinal, como a atenuação, a reflexão, a refração, a dispersão, a difração e o desvanecimento.

As linhas de transmissão ou guias de ondas são responsáveis pelo transporte da energia eletromagnética desde o transmissor até à antena, ou desde a antena até ao recetor (Balanis, 2005). Existem diferentes tipos de guias de ondas, entre os quais a linha bifilar, o cabo coaxial, a linha impressa e o guia metálico.

Uma antena é uma estrutura de transição entre o espaço livre e o guia de ondas (Balanis, 2005), que pode ser utilizada de forma singular ou em agregado, dependendo do propósito do sistema a desenvolver. Os dipolos e monopolos são exemplos de antenas

²² Do inglês: *International Telecommunications Union – Radiocommunication Sector*

que são normalmente constituídas por um fio condutor, mas que também podem ser feitos em circuito impresso.

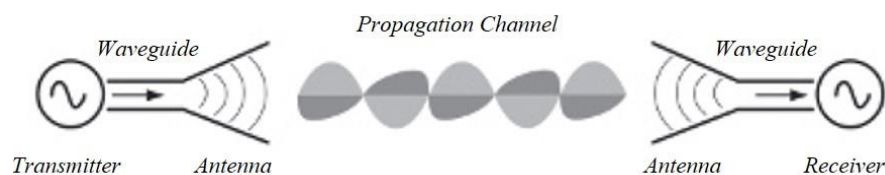


Figura 3.1: Esquema básico de um sistema de radiocomunicações²³

3.2 Definição dos Objetivos

Um veículo não tripulado tem duas formas de operar: remotamente, o que implica a existência de uma linha de comunicação entre o VENT-SUP e um centro de comando e controlo exterior ao veículo, através da qual existe a partilha de informação necessária à sua atividade; ou de forma autónoma, situação em que não existe controlo exterior sobre o veículo.

A impossibilidade de estabelecer uma linha de comunicação física entre o veículo de superfície e a estação de terra, obriga-nos a recorrer a comunicações sem fios. A escolha de comunicações por radiofrequência tem de obedecer a alguns requisitos, tais como a distância a que se pretende comunicar, a quantidade e o tipo de dados a enviar, a plataforma em que vai ser instalada, as restrições governamentais ou a capacidade de implementação.

Tendo em conta as principais operações que VENT-SUP irá desempenhar, é essencial que este garanta “Apoio ao comando, controlo e comunicações (C3) através de relé e transporte de dados” (Estado Maior da Armada, 2016). Como tal, é necessário que o veículo possua a capacidade de enviar uma grande quantidade de dados, o que implica uma menor distância de comunicação. Por outro lado, é ideal que o veículo comunique a

²³ Adaptado de (Huang & Boyle, 2008)

uma grande distância, mas a essa realidade está associada uma diminuição na capacidade de transferência de dados.

Para garantir esta dupla necessidade, será necessário implementar duas bandas de frequência que cumpram os pré-requisitos. Uma das bandas terá como objetivo comunicar a longas distâncias, ou seja, um canal de telemetria usado para enviar e receber informações básicas de sobrevivência do veículo. Para isso podemos usar o início da banda UHF (cerca de 400 MHz), o que nos pode conferir alguns quilômetros de distância de comunicação, mas uma largura de banda pequena. Definimos então para esta banda, numa fase inicial, uma distância objetivo de 10km (cerca de 5 milhas) e uma largura de banda suficiente para transmitir e receber informações de dois ou três sensores. A segunda banda terá como objetivo transmitir e receber grandes quantidades de dados. Para isso podemos usar o fim da banda UHF (cerca de 2,5 GHz), o que nos confere uma maior largura de banda, mas uma distância de comunicação menor. Para esta banda, foi então inicialmente definida uma largura de banda suficiente para fazer *streaming* de vídeo, a mais de 50 kbps²⁴ (704×480@1fps).

Após análise dos regulamentos nacionais implementados pela ANACOM²⁵, verificou-se que era possível cumprir com o disposto acima, recorrendo a duas bandas de frequência ISM²⁶: 433,05-434,79 MHz e 2,4-2,4835 GHz (Autoridade Nacional de Comunicações, 2010). Uma vez conhecidas as bandas de frequência a utilizar e considerando as características da plataforma em que vão ser implementadas, foi escolhido um monopolo clássico de um quarto de comprimento de onda para a primeira banda e um monopolo impresso, também de um quarto do comprimento de onda, para a segunda. A primeira banda tem uma menor largura de banda (1,74 MHz = 0,40%) mas proporciona um alcance maior que a segunda (que tem largura de banda 83,5 MHz = 3,42%).

²⁴ Streaming de vídeo de baixa qualidade (704x480) e baixa taxa de transferência, 1 *frame* por segundo <http://stardot.com/bandwidth-and-storage-calculator>

²⁵ Autoridade Nacional de Comunicações

²⁶ Do inglês: *industrial, scientific and medical*. São bandas de frequências, que não necessitam licenciamento, especialmente alocadas para propósitos científicos, industriais e médicos.

Ao longo deste capítulo serão explicados de forma aprofundada o desenvolvimento das antenas selecionadas, todos os elementos escolhidos que compõem o sistema de telecomunicações e os conhecimentos inerentes ao funcionamento do sistema.

Como mencionado anteriormente, existem vários tipos de antenas, sendo os dipolos e os monopolos as antenas mais usadas e de estudo e construção mais simples.

Uma antena do tipo monopolo é metade de uma antena dipolar, montada sobre um plano de terra (Collin, 1985), como se pode verificar na figura 3.2, e é caracterizada por ter uma fácil implementação. Comparativamente ao dipolo, pode afirmar-se que o monopolo correspondente apresenta metade da impedância de entrada e o dobro da diretividade, tornando-se, desta forma, uma solução mais conveniente para certas aplicações. Os monopolos de $1/4\lambda$ (um quarto do comprimento de onda) são os mais comuns devido a estarem na primeira ressonância, ou seja, terem o menor comprimento que apresenta uma impedância de entrada resistiva, não necessitando normalmente de sistema de adaptação de impedâncias.

Como referido anteriormente, vamos escolher um monopolo de fios para a banda de frequências mais baixa e um monopolo em circuito impresso para a banda mais alta.

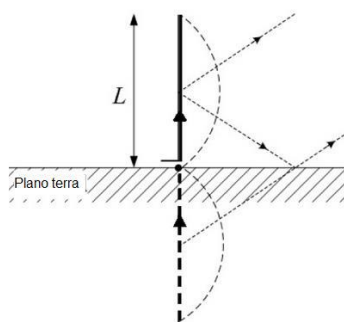


Figura 3.2: Monopolo vertical sobre um plano de terra infinito²⁷

Os monopolos apresentam um diagrama de radiação omnidirecional (radiação constante no plano perpendicular à direção definida pelo monopolo). Sendo o monopolo

²⁷ Adaptado de (L. Liu, S.W. Cheung, 2012)

montado orientado verticalmente, o que garante invariância da comunicação a movimentos de rotação do veículo. Como o diagrama de radiação é pouco diretivo no plano vertical, a sensibilidade a oscilações causadas pela ondulação também é pequena.

O estudo que será feito no presente capítulo é um guia geral que pode ser usado para a construção de antenas noutras bandas que não aquelas que foram usadas, e para as quais pode ser difícil encontrar antenas disponíveis no mercado.

3.3 Monopolo de Fios

3.3.1 Dimensionamento e Otimização

3.3.1.1 Monopolo Ideal

Para fazer o dimensionamento do monopolo é necessário conhecer quais as suas principais características e qual o comportamento de determinados parâmetros dos monopolos ideais. A frequência escolhida para o estudo foi de 433,92 MHz, uma vez que representa o centro da banda ISM (433,05-434,79 MHz), à qual corresponde um comprimento de onda (λ) de 69,14 cm, dando assim origem a um monopolo ($\lambda/4$) com um comprimento de 17,28 cm. Este monopolo é demasiado grande para ser implementado em tecnologia de circuito impresso pois, depois de montado no topo do mastro de uma embarcação, apresentaria uma resistência ao vento demasiado elevada. Para fazer o estudo da antena foi utilizado o programa *4NEC2*²⁸, um programa de visualização, modelação e optimização de antenas.

O software *4NEC2* utiliza o método dos momentos (Huang & Boyle, 2008) para fazer o cálculo dos diagramas de radiação e parâmetros que definem uma antena de fio. Uma estrutura de antena simples, um fio, pode ser dividida em vários segmentos, sendo que a corrente de cada segmento pode ser calculada. O “momento” é numericamente a dimensão da corrente multiplicada pelo vetor que descreve o segmento, completando assim um diagrama da distribuição da corrente junto à antena (Collin, 1985).

²⁸ Acesso: <https://www.qsl.net/4nec2/>

Tendo em vista a obtenção do primeiro modelo do monopolo, foram estabelecidas as seguintes condições iniciais: monopolo de $1/4 \lambda$, com 17,28 cm de altura, raio igual a 1 milímetro, fio condutor perfeito com 15 segmentos (explicado a baixo), potência de entrada de 100 mW, impedância de entrada de 50 ohms, alimentado na base, com um plano de terra perfeito. Podem observar-se na figura 3.3 os principais parâmetros do monopolo e a sua geometria e o seu diagrama de radiação, na figura 3.4.

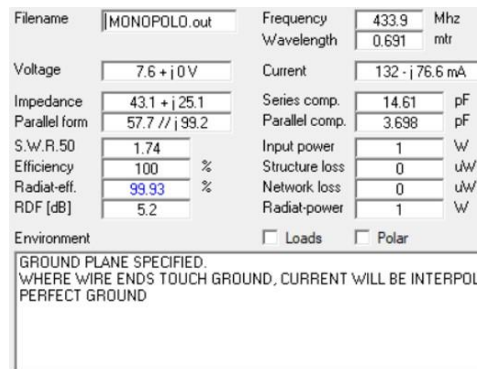


Figura 3.3: Parâmetros obtidos na primeira simulação (condições iniciais)

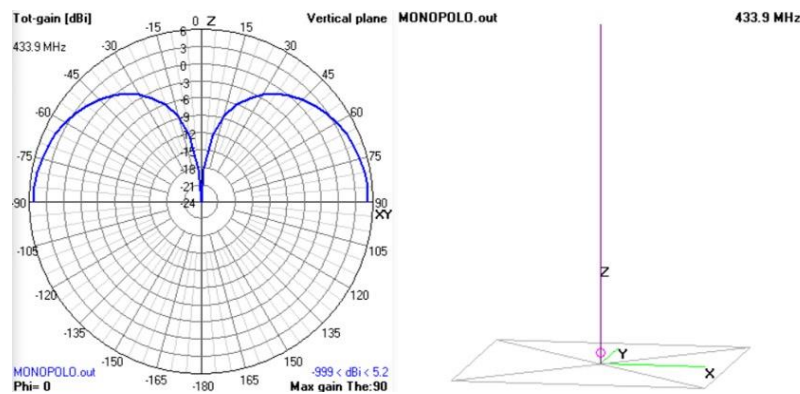


Figura 3.4: Diagrama de radiação e geometria do monopolo (condições iniciais)

Como referido anteriormente, o programa *4NEC2* utiliza o método dos momentos para o cálculo dos parâmetros do monopolo e, para isso, segmenta a geometria da antena para fazer as diferentes simulações. De modo a encontrar o número ideal de segmentos, foram utilizados critérios descritos nos tutoriais do programa e foi feito também um estudo de

convergência para garantir os melhores resultados e também estabilidade numérica. Os requisitos computacionais, segundo Knott (Knott, 2009), são cumpridos, para o caso em estudo, se o monopolo tiver mais de 5 segmentos. Após a análise do comportamento dos diferentes parâmetros da antena com a variação do número de segmentos (gráficos 3.1, 3.2 e 3.3), conclui-se que o número de segmentos pode variar entre 5 e 52, zona de estabilidade comum para todos os parâmetros.

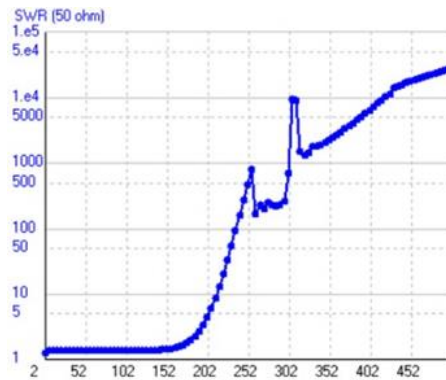


Gráfico 3.1: Comportamento da relação de onda estacionária com a variação do número de segmentos

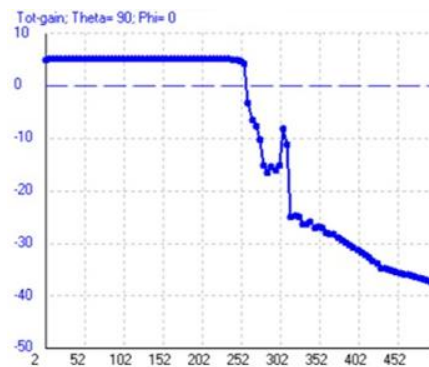


Gráfico 3.2: Comportamento do ganho com a variação do número de segmentos

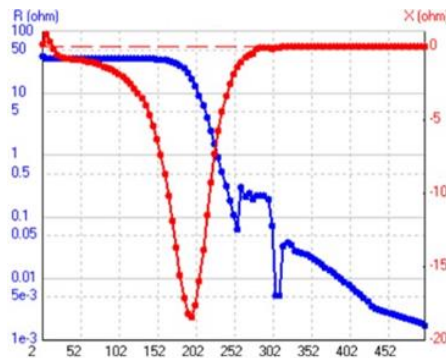


Gráfico 3.3: Comportamento da impedância com a variação do número de segmentos

Aquando o teste de convergência o programa apresentou uma mensagem de erro quando se utilizava mais de 30 segmentos. Para aliar a rapidez de cálculo com a precisão dos valores obtidos definiu-se como 15 o número de segmentos a utilizar.

A impedância (Z) é dada pela expressão: $Z = R + jX$, onde “R” representa a resistência e “X” a reatância. Pode verificar-se na figura 3.3 que a impedância da antena simulada é dada por: $Z = 43,1 + j25,1$, não sendo este um valor satisfatório. Para que a antena seja ressonante na frequência central, a reatância terá de ser nula e a resistência igual à impedância fornecida pelo circuito de alimentação da antena. Dado que a antena em estudo irá ser alimentada por um cabo coaxial com 50 ohms de impedância característica, o valor ideal da impedância da antena será dado por: $Z = 50 + j0$. Para se obter os valores padrão para esta antena, terá de se otimizar a sua altura e raio, conseguindo-se assim uma antena ressonante na frequência central e uma largura de banda maximizada. Estes valores são retirados de gráficos como o representado na figura 3.5.

Na tabela seguinte encontram-se os valores calculados nas várias simulações para a frequência central da banda - 433,92 MHz.

Raio [mm]	Otimizado	Altura [m]	Ganho [dBi]	Reatância [Ω]	Resistência [Ω]	ROE	Largura de Banda [MHz] ($ROE \leq 2$)
1	Não	0,1728	5,2	25,1	43,1	1,74	52

	Sim	0,1628	5,14	-0,1	35,9	1,39	56
2	Não	0,1728	5,21	25,7	45,2	1,72	61
	Sim	0,1674	5,14	0,15	36	1,39	68
3	Não	0,1728	5,22	25,6	46,9	1,69	69
	Sim	0,1593	5,14	0,06	36,1	1,39	77
4	Não	0,1728	5,22	25,2	48,4	1,66	76
	Sim	0,1582	5,14	0,02	36,2	1,38	88
5	Não	0,1728	5,23	24,4	49,7	1,62	83
	Sim	0,1575	5,15	0,1	36,4	1,37	96

Tabela 3.1: Otimização dos parâmetros do monopolo ideal variando o raio e altura

Os valores dos parâmetros calculados nas simulações são os esperados, pois quanto maior o raio de uma antena, maior a sua largura de banda.

Segundo Collin, um monopolo de fio de um quarto de comprimento de onda ideal, montada sobre um plano de terra perfeito, tem uma resistência de radiação de 36,56 ohms, metade da do dipolo de meia-onda (Collin, 1985). Tendo em conta a tabela 3.1, otimização dos parâmetros com a variação da altura e do raio, depreende-se que os resultados que conferem melhores características ao monopolo são obtidos quando o raio é igual a 4 mm, pois obtém-se reatância próxima de zero e resistência mais perto dos 50 Ω . Apesar dos valores obtidos com um raio de 4 mm serem os mais favoráveis, o valor de raio escolhido para o monopolo foi de 2 mm pois, por razões a explicar no próximo capítulo, o plano de terra influencia também o raio do condutor principal. Assim sendo, as características do condutor principal do monopolo serão:

- Dimensões: 16,74 cm de altura e 2 mm de raio
- Ganho: 5,14 dBi
- Impedância: $(36 + 0,15j) \Omega$
- Largura de banda: 68 MHz

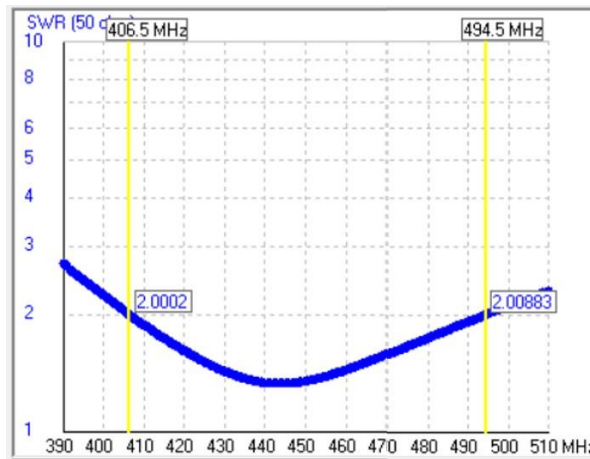


Figura 3.5: Obtenção da largura de banda através da variação da ROE em frequência

O monopolo ideal, obtido através do dimensionamento anteriormente descrito foi concebido num cenário utópico, ou seja, com um plano de terra infinito e construído a partir de um fio condutor perfeito. No entanto, na impossibilidade da existência de condições perfeitas, é necessário estudar o dimensionamento do plano de terra e ter em consideração o material de construção da antena, pela influência que tem no seu comportamento. O material escolhido para a construção da antena foi o cobre, uma vez que é um material condutor comum, que apresenta valores de reatância baixos tendo em conta o nível de perdas na estrutura.

3.3.1.2 Plano de Terra

O plano de terra perfeito consiste num plano infinito constituído por material condutor perfeito, o que permite obter um diagrama de radiação ideal. Na figura 3.6, é demonstrada a diferença entre um plano de terra perfeito e um plano de terra de água salgada ($\epsilon_r = 81$; $\sigma = 5 \text{ mS/m}$), onde as principais diferenças residem no ganho da antena e na direção de maior ganho, porque o plano de água salgada provoca atenuações, que se devem ao facto de não apresentar os valores ideais de condutividade elétrica e de constante dielétrica.

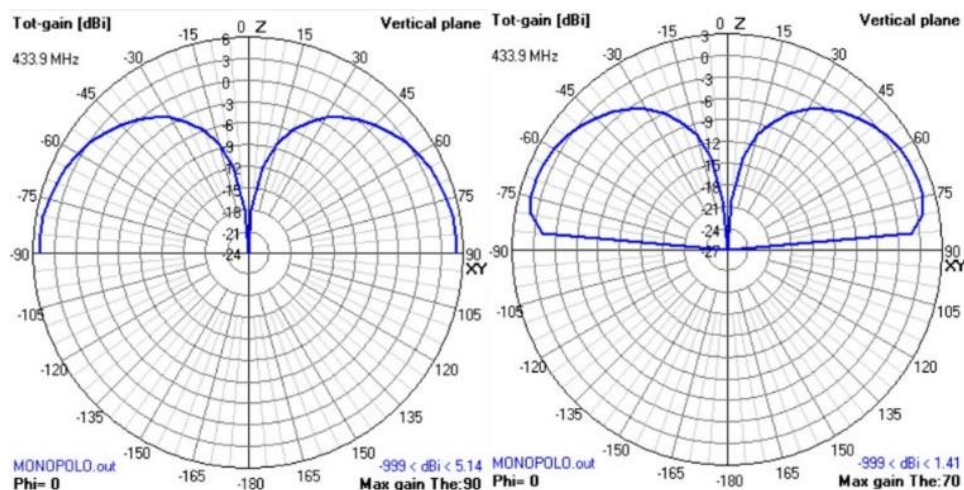


Figura 3.6: Diagramas de radiação de uma antena com plano de terra infinito: condutor perfeito (à esquerda) e água salgada (à direita) ($f = 433,92$ MHz)

Com vista a simular um plano de terra que satisfaça os requisitos do projeto é importante saber quais os valores dos parâmetros que devemos obter nos diferentes testes. Para isso foi criado um plano de terra que, de certa forma, fosse equivalente ao plano ideal. Este plano, representado na figura 3.7, é constituído por uma malha circular com 50 centímetros de raio e uma aba cônica com o maior raio de 80 centímetros e com 45 graus de ângulo em relação à vertical, sendo os dois planos uma simulação de uma superfície sólida, pois em radiofrequência qualquer espaço com uma dimensão inferior a $\lambda/20$ pode ser considerado sólido²⁹.

²⁹ *Antenna Modeling With 4NEC2*, Craig Weixelbaum, 2014

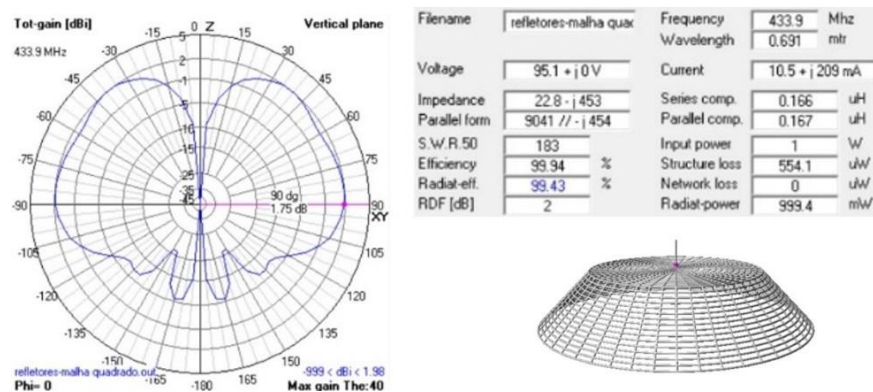


Figura 3.7: Simulação de um plano de terra electromagneticamente infinito

Através do diagrama de radiação obtido conclui-se que o plano de terra não é exequível nos termos deste projeto, pois apesar de se conseguir eliminar praticamente a radiação nas direções superiores a 90 graus, o ganho e dimensões deste plano não favorecem a sua implementação, tendo assim de se recorrer a uma solução mais simples, que não apresente aspetos impeditivos. A ausência de um plano de terra pode ser compensada pela instalação de um plano constituído por refletores com um tamanho aproximado de 0,3 vezes o comprimento de onda. Estes refletores simulam um plano de terra suficiente para que impedância e o ganho do monopolo seja aproximadamente equivalente a um dipolo de meio comprimento de onda (Collin, 1985).

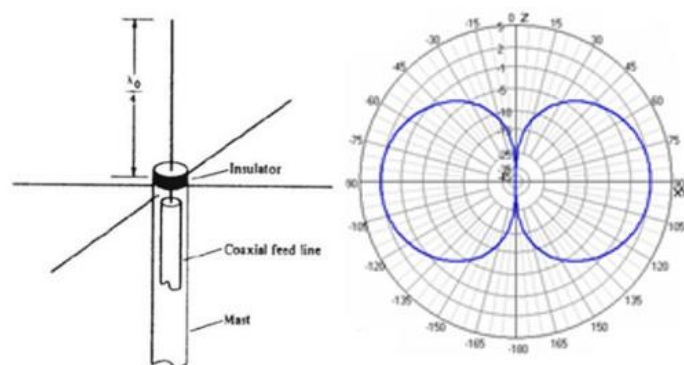


Figura 3.8: Monopolo com plano de terra constituído por 4 refletores e respetivo diagrama de radiação³⁰

³⁰ Adaptado de (Collin, 1985)

Para simular o plano de terra foi utilizado o conceito acima referido: o uso de refletores constituídos pelo mesmo material que o monopolo, colocados na sua base. De modo a otimizar o monopolo foram feitas várias experiências variando o número, a dimensão e a inclinação dos refletores em relação à horizontal, pois obtém-se melhores resultados do que no caso em que se altera a dimensão do monopolo e se mantem a dimensão dos refletores.

Como mencionado no capítulo anterior, o raio do condutor principal e dos refletores é um aspeto que limita as opções no momento da escolha de qual as melhores características para se obter o monopolo com parâmetros que satisfazem as nossas necessidades. Este problema surge devido à acumulação de cargas que ocorre na base do monopolo quando se coloca um plano de terra constituído por refletores, influenciando assim o ganho e diagramas de radiação da antena.

Este problema foi detetado no momento da simulação da dimensão e características dos refletores, pois o software *4NEC2* indicava erro no cálculo entre a transição do condutor principal para os refletores. Após várias simulações, concluiu-se que o condutor principal não poderia exceder os 3 mm e que os refletores não podem também exceder essa medida, caso contrário, os parâmetros obtidos estariam a ser prejudicados pelas cargas acumuladas, não obtendo assim o máximo rendimento da antena em projeto.

Para saber qual a melhor configuração dos refletores relativamente ao monopolo, fizeram-se várias simulações variando três características dos refletores: número de refletores (4, 6 ou 8); comprimento otimizado (de forma a ajustar a impedância), maximizando o valor da resistência (objetivo: 50 Ω) e minimizando o valor da reatância (objetivo: 0 Ω); inclinação (0°, 30°, 45° ou 60°). Os resultados obtidos encontram-se na tabela 3.2.

Número de refletores	Inclinação [°]	Dimensão otimizada [cm]	Ganho [dB]	Direção de maior ganho [°]	Impedância [Ω]	ROE	Eficiência de radiação [%]
4	0	21,26	0,75	90	24,8 – j 0,01	2,01	82,89
	30	17,91	1,77	90	42,7 + j 0,02	1,17	93,6
	45	14,11	2,09	90	48,4 + j 0,02	1,03	97,8
	60	12,89	2,28	90	53 – j 0,06	1,06	98,3
6	0	24,02	0,73	85	25,1 – j 0,01	1,99	76,51
	30	18,85	1,81	90	41,4 – j 0,02	1,21	86,9
	45	16,95	2,1	90	46,7 – j 0,005	1,07	90,5

	60	15,13	2,99	90	$48,1 + j 0,02$	1,04	99,1
8	0	22,34	0,71	85	$25,3 - j 0,007$	1,98	76,86
	30	14,09	1,82	90	$40,7 + j 0,01$	1,23	87,9
	45	12,24	2,1	90	$45,8 - j 0,02$	1,09	91,4
	60	11,38	2,73	90	$47,4 + j 0,03$	1,05	99,6

Tabela 3.2: Estudo do plano de terra ($f = 433,92$ MHz)

Conclui-se assim o estudo do monopolo clássico, obtendo-se uma antena com 3 dBi de ganho, 68 MHz de largura de banda, 16,82 cm de comprimentos e 2 mm de raio, com um plano de terra constituído por 6 refletores a 60° relativamente ao plano horizontal, com 15,13 cm de comprimento.

3.3.1.3 Monopolo Otimizado

O dimensionamento e otimização da estrutura da antena teve como base a análise e manipulação dos parâmetros de um monopolo, de modo a obter uma estrutura que possibilite o fluxo de informação entre a estação de terra e o veículo.

Após a escolha da frequência de trabalho, que teve em conta os requisitos do veículo e as bandas disponíveis, foi necessário escolher o tipo de antena a utilizar. A solução escolhida foi o monopolo de $1/4\lambda$ que sofreu um processo de otimização até ser encontrada a estrutura que melhor desempenha as funções desejadas.

Inicialmente, foi criado um monopolo com características padrão em condições ideais. Através da análise dos resultados obtidos numericamente, foram manipulados vários parâmetros da antena até ser encontrado o monopolo com as características ideais. Conhecido o comportamento da antena em condições perfeitas, deu-se início à modulação do plano de terra, tendo como objetivo a construção de uma estrutura que se aproximasse ao máximo das condições inicialmente previstas. Depois de várias iterações no processo de otimização do monopolo, obteve-se uma estrutura cujos parâmetros se encontram descritos na tabela 3.3.

Frequência	Comprimento de Onda	Potência de Entrada	Voltagem	Corrente	Potência Radiada
------------	---------------------	---------------------	----------	----------	------------------

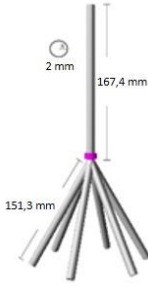
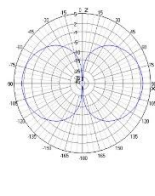
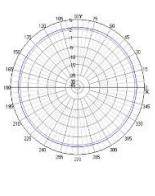
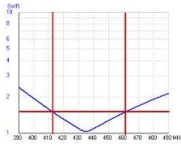
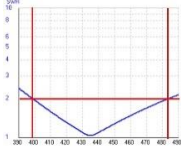
433,91 MHz	69,14 cm	100 mW	2,09 V	47,7 – j 4 mA	99,95 mW
Perdas na Estrutura	Impedância	ROE (mínima)	Eficiência	Eficiência de Radiação	Ganho [dBi]
44,86 μ W	48,1 + j 0,02 Ω	1,04	99,96 %	99,1 %	2,99 (90°)
Geometria		Diagrama de Radiação		Largura de Banda	
		Plano Vertical	Plano Horizontal	$P_{\text{Refletida}} = 4\%$	$P_{\text{Refletida}} = 11\%$
				 38 MHz	 69 MHz

Tabela 3.3: Quadro resumo dos resultados obtidos

3.3.2 Construção e Testes

De forma a concluir o projeto do monopolo, foi necessário conceber um esquema de construção para o mesmo, com o objetivo de o tornar aplicável ao sistema de telecomunicações. O principal desafio foi equacionar a montagem do conector de alimentação da antena em conjunto com os refletores e de que forma se poderiam colocar os mesmos. A solução encontrada foi o desenho de uma base que interligasse o plano de terra à base do conector SMA e o condutor do mesmo conector ao condutor principal da antena, como é demonstrado na figura 3.9.

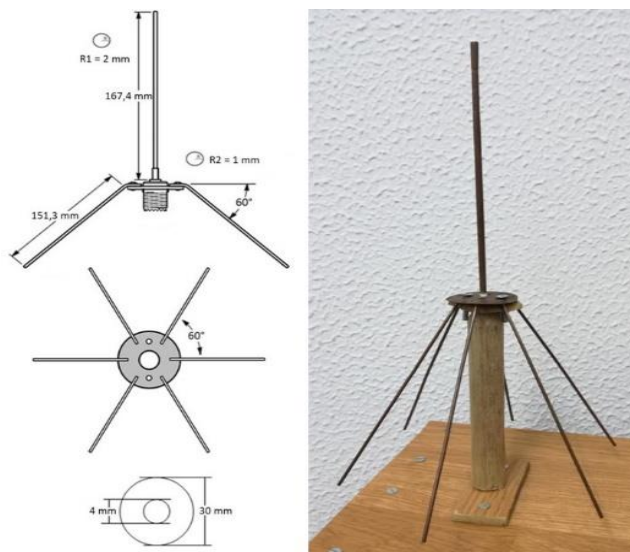


Figura 3.9: Esquema de construção e respetivo resultado (monopolo de fios)

Dado o esquema de montagem do circuito da antena, foi necessário adquirir o conector com o espigão capaz de ser inserido no seio da secção da antena e um disco que unisse todos os refletores. O condutor principal foi obtido a partir de uma vara de cobre de 10 mm, tendo-se torneado a peça até ter a dimensão pretendida (figura 3.10).



Figura 3.10: Torno utilizado para moldagem do monopolo

O plano de terra foi obtido a partir da solda a estanho entre um disco de cobre com os fios de 2 mm de diâmetro. Tendo todas as partes constituintes prontas, faltava unicamente unir todas elas ao conector SMA. O espigão foi inserido e soldado ao condutor principal

e a base do conector foi aparafusada ao disco do plano de terra, concluindo assim a montagem do monopolo clássico.

Uma vez terminada a construção do monopolo, passou-se ao teste do seu desempenho, tendo este sido feito em dois passos: O primeiro foi a medição do coeficiente de reflexão, através de um analisador de redes vetorial (*Agilent Technologies E5071C*) do Instituto de Telecomunicações, Instituto Superior Técnico (figura 3.11); e a segunda medição foi feita através de um analisador de espectros.

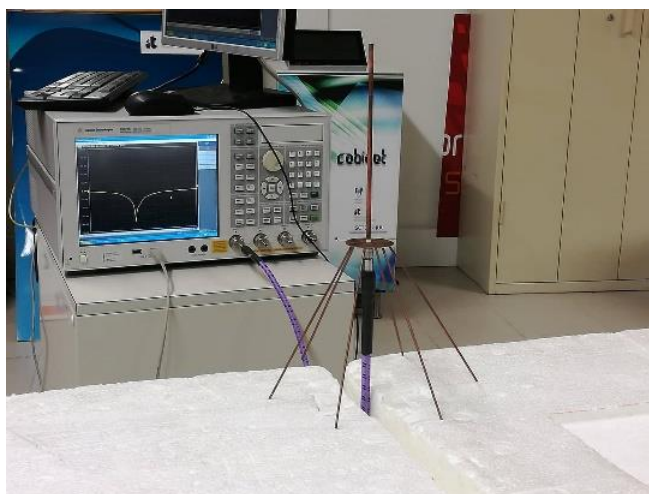


Figura 3.11: Medição do coeficiente de reflexão

A primeira medição, coeficiente de reflexão ($S_{1,1}$), permite-nos saber se a antena está adaptada para a frequência central pretendida, a largura de banda e a percentagem de potência refletida. No primeiro teste, obteve-se uma curva de reflexão satisfatória em termos de forma, mas não se encontrava na frequência pretendida. Para colmatar essa situação, foi necessário encurtar a dimensão dos refletores de forma a aumentar a frequência central da antena. Este processo foi repetido até serem atingidos valores mais próximos dos esperados, sem prejudicar o desempenho geral da antena. Os resultados experimentais finais encontram-se no gráfico 3.4, não tendo sido assim possível atingir a frequência central desejada, mas ficando com um fator de reflexão nos 433 MHz de, aproximadamente, -23 dB.

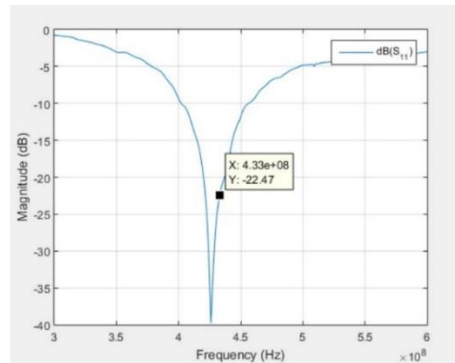


Gráfico 3.4: Curva do fator de reflexão do monopolo clássico

3.4 Monopolo Impresso

Uma antena impressa é composta por um elemento metálico capaz de emitir radiação, impresso sobre um substrato dielétrico que, do lado oposto, possui um plano de terra. O elemento impresso é, regra geral, de cobre ou ouro e passível de ter qualquer forma, sendo normalmente usadas formas canônicas para facilitar o seu estudo e análise. Idealmente, o substrato deve possuir uma constante dielétrica com baixo valor ($\epsilon_r < 2,5$), de forma a obter maior rendimento e maior largura de banda (Garg, Bhartia, Bahl, & Ittipiboon, 2001).

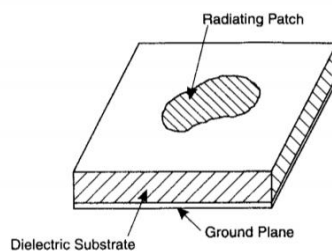


Figura 3.12: Configuração de uma antena impressa³¹

Faz também parte da constituição de uma antena impressa o método para efetuar a sua alimentação. Segundo (Sharma, 2017), existem quatro métodos principais. A linha

³¹ Retirado de (Garg et al., 2001)

microfita (a) é um deles. Neste tipo de técnica é usada uma linha de material condutor para fazer ligação entre o elemento impresso e o limite exterior do substrato, de forma a permitir a ligação para um conector, tendo a vantagem de não haver a necessidade de adaptação de impedâncias. Sonda coaxial (b) é também uma das técnicas mais utilizadas para a alimentação de antenas impressas, sendo o condutor interno do cabo coaxial soldado ao elemento impresso, atravessando a camada de dielétrico, e o condutor externo do cabo soldado ao plano de terra. A técnica de acoplamento eletromagnético por abertura (c) utiliza uma linha de alimentação separada do elemento impresso pelo plano de terra, havendo uma abertura neste plano para realizar alimentação da antena. Acoplamento eletromagnético por proximidade (d), também conhecida como adaptação eletromagnética, é uma técnica em que a alimentação é feita através da proximidade entre a linha de alimentação e o elemento impresso. São utilizados dois substratos, entre os quais atravessa a linha de alimentação, estando o elemento impresso sobre o substrato superior. A principal vantagem desta técnica é a eliminação das perdas na linha de alimentação, aumentando assim o rendimento.

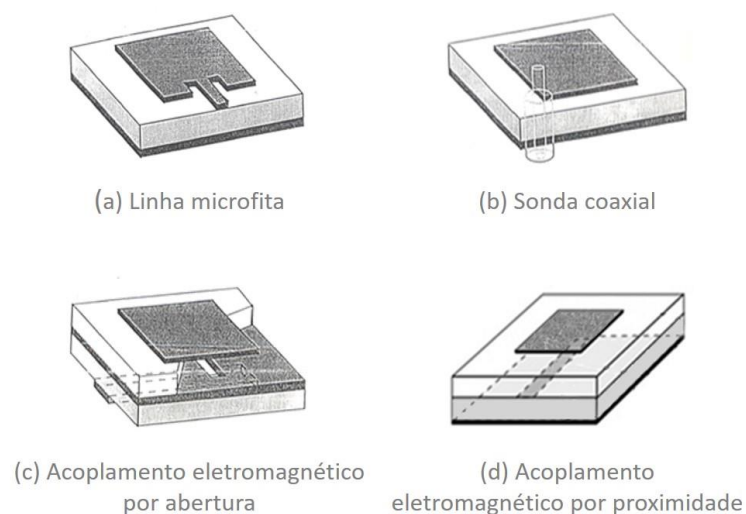


Figura 3.13: Tipos de alimentação de uma antena impressa³²

As antenas impressas apresentam importantes vantagens, tais como: baixo perfil, peso e volume; baixo custo de fabrico; são passíveis de integração de circuitos ativos de

³² Adaptado de (Barrou, El Amri, & Reha, 2017) e (Sharma, 2017)

microondas; grande flexibilidade no que concerne a largura de banda, diagrama de radiação e polarização. Como principais limitações são de realçar: largura de banda estreita; ganho relativamente baixo; dificuldade em atingir pureza de polarização. As especificações e características deste tipo de antenas permitem uma vasta aplicabilidade no mundo das comunicações, tais como nas comunicações móveis, comunicações via satélite, biomedicinas, antenas ativas, antenas inteligentes, entre outras.

3.4.1 Dimensionamento e Otimização

O processo de desenvolvimento de uma antena impressa obedece a três passos fundamentais: pré-dimensionamento, projeto e fabrico. Esta secção irá abordar os dois primeiros passos, sendo o terceiro abordado na secção seguinte.

O pré-dimensionamento de uma antena impressa pode ser efetuada de duas formas diferentes: modelo da linha de transmissão e modelo da cavidade ressonante. Para calcular a dimensão do elemento impresso é necessário, primeiramente, fazer a escolha do substrato de forma a determinar a constante dielétrica e a altura do substrato. Para o projeto em questão foi utilizado o substrato “*ROGERS duroid 5880*” com as seguintes características:

- Constante dielétrica relativa: $\epsilon_r = 2,20$;
- Tangente de perdas: $\tan \delta = 0,0006$;
- Espessura: $h = 1,575 \text{ mm}$.

Após a escolha do substrato, e conhecidas as suas características, pode-se determinar o comprimento (L) do elemento impresso a partir das equações apresentadas de seguida, sendo que para a largura (W) foi utilizado 1 mm e comprimento de onda (λ_0) 12,24 mm, pois a frequência central é aproximadamente 2,45 GHz.

$$1) \quad \epsilon_{re} = \frac{\epsilon_r + 1}{2} + \left(\frac{\epsilon_r - 1}{2} \right) \times \left(1 + \frac{12h}{w} \right)^{-\frac{1}{2}} = 1,7345$$

$$2) \quad \lambda_g = \frac{\lambda_0}{\sqrt{\epsilon_{re}}} = 92,98 \text{ mm}$$

$$3) \quad L_e = \frac{\lambda_g}{4} = 23,24 \text{ mm}$$

$$4) \Delta L = 0,412 \times h \times \frac{(\epsilon_{re}+0,3) \times \left(\frac{W}{h}+0,264\right)}{(\epsilon_{re}-0,258) \times \left(\frac{W}{h}+0,8\right)} = 0,56 \text{ mm}$$

$$5) L = L_e - 2 \times \Delta L = 22,12 \text{ mm}$$

Concluído o primeiro passo do processo de desenvolvimento de uma antena impressa (Peixeiro, 2017), o pré-dimensionamento, podemos passar para o projeto da antena recorrendo a ferramentas de desenho, simulação e análise de estruturas eletromagnéticas. Para este efeito foi utilizado o *CST MICROWAVE STUDIO*³³, programa utilizado pelo Instituto Superior Técnico, que reúne todas as ferramentas necessárias para fazer o projeto de qualquer tipo de antena. Esta ferramenta utiliza várias técnicas de cálculo, tais como: *finite integration technique*, *Time Domain Transmission Line Matrix*, *Frequency Domain Solver*, *Integral Equation Solver*, *Multilayer Solver*, *Asymptotic Solver* e *Eigenmode Solver*. Todas estas técnicas são utilizadas pelo *CST*, sendo que a principal diferença entre elas é o tipo de malha que utilizam, que varia consoante a complexidade da estrutura, para fazer o cálculo dos diferentes parâmetros para análise.

Para o projeto da antena foram utilizadas as medidas apresentadas na figura 3.14, baseadas no pré-dimensionamento, tendo sido usadas para dimensões de substrato as medidas $W_g = 100 \text{ mm}$ (largura do substrato) e $L_g = 70 \text{ mm}$ (comprimento do substrato).

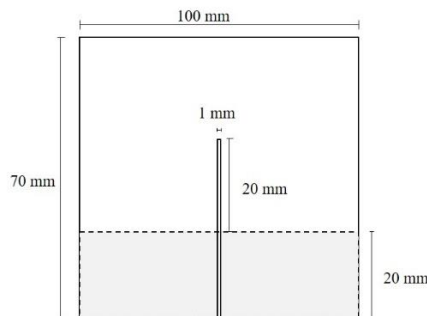


Figura 3.14: Medidas iniciais utilizadas no software *CST*

De forma a possibilitar a simulação foi necessário projetar o conector a ser usado mais tarde, conector SMA, com uma impedância de 50Ω apresentado na figura 3.15. O conector é constituído por cobre e teflon, sendo este ligado mecânica e eletricamente ao

³³ Acesso: <https://www.cst.com/products/cstmws>

elemento impresso (monopolo) e plano de terra. A antena é constituída pelo elemento impresso de um dos lados do substrato e o plano de terra do lado oposto. Esta será alimentada por um cabo coaxial de $50\ \Omega$, alimentando a antena a partir do conector SMA.

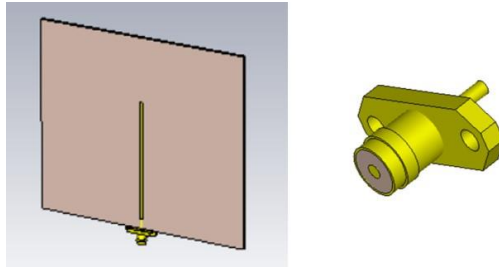


Figura 3.15: Monopolo impresso e conector SMA projetado no software CST

À semelhança do dimensionamento do monopolo de fios, os parâmetros que são tomados em conta são, fundamentalmente, o coeficiente de reflexão, ganho e diagrama de radiação. Após a introdução no software o esquema desejado para a antena impressa, obtiveram-se resultados que não cumpriam os requisitos de uma antena adaptada, pois o coeficiente de reflexão mínimo encontrava-se nos 2,6 GHz com uma amplitude de -11 dB e um diagrama de radiação com mínimos no plano horizontal, onde é objetivo ter o máximo valor de radiação.

De forma a otimizar os parâmetros obtidos no software foram alteradas as medidas iniciais, altura do monopolo, altura do plano de terra e largura e altura do substrato, tendo em atenção a alteração dos parâmetros, na tabela 3.4 estão os resultados obtidos.

	Dimensões [mm]				S _{1,1}		Ganho [dBi] (f = 2,45 GHz)
	Wg	Lg	L	Fi	Mag. [dB]	Freq. [GHz]	
1	100	70	20	20	-11	2,58	3,6
2	110	70	20	20	-13	2,35	3,41
3	110	70	25	20	-11	2,4	3,41
4	110	80	25	20	-8,5	2,42	3,42
5	110	80	25	25	-25	3	3,32
6	110	80	25	30	-11	3	3,22
7	110	90	25	30	-18	2,8	3,22
8	110	90	25	25	-15	2,8	3,31
9	110	95	25	25	-15	2,63	3,31
10	110	95	30	25	-13	2,21	3,32
11	110	95	30	30	-26	2,5	3,23
12	105	95	30	30	-52	2,4	3,15
13	105	90	30	30	-19	2,4 – 2,65	3,16
14	100	90	30	30	-27	2,45	3
15	100	90	32	30	-48	2,45	2,99

16	110	90	32	30	-23	2,6	3,23
17	110	100	32	30	-22	2,35	3,24
18	110	95	32	30	-33	2,4	3,23
19	110	95	32	25	-12	2,5	3,32
20	110	95	32	35	-13	2,6	3,15
21	110	95	31	30	-27	2,41	3,23
22	110	95	31,5	30	-30	2,4	3,23
23	115	95	31,5	30	-35	2,5	3,26
24	110	100	31,5	30	-12	2,27	3,27
25	110	95	31,5	29	-34	2,45	3,25

Tabela 3.4: Otimização dos parâmetros do monopolo impresso

Com o processo de otimização do monopolo foi possível obter uma antena adaptada e com parâmetros adequados à aplicação que se pretende implementar. De forma a obter o maior rendimento possível da antena, é necessário obter um coeficiente de reflexão muito menor que -10 dB na frequência central pretendida (2,45 GHz). Os resultados obtidos para o caso ótimo ($W_g = 110$ mm, $L_g = 95$ mm e $L = 31,5$ mm) estão no gráfico 3.5 e figura 3.18. No gráfico de $|S_{11}|$ podemos ver que na frequência central temos um coeficiente de reflexão de -34 dB, o que nos indica uma percentagem de potência refletida inferior a 0,1% e uma largura de banda (para $|S_{11}| < -10$ dB) de 620 MHz.

Para obter um diagrama de radiação adequado à aplicação, o ganho não poderá ser tão alto como o obtido inicialmente. Com a variação das dimensões do monopolo impresso foi possível obter um ganho de 3,25 dBi na direção de maior ganho, que se encontra 28 graus acima do plano horizontal.

O software utilizado permite obter diagramas de radiação nos planos, $\theta = 90$, $\phi = 90$ e $\phi = 0$. A partir da análise dos diagramas de radiação nestes planos podemos concluir que a antena radia pouca energia para setores abaixo do plano horizontal, aumentando assim o seu ganho nos setores pretendidos, obtendo o seu máximo 28 graus acima do plano horizontal. Radia de forma quase omnidirecional.

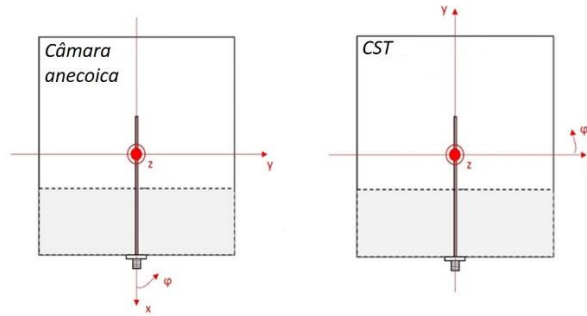


Figura 3.16: Sistema de eixos (XYZ) e ângulos de *Theta* e *Phi* usados

Conclui-se assim o projeto do monopolo impresso, obtendo-se uma antena completamente adaptada e com os seus parâmetros otimizados. De todo o processo de implementação de uma antena impressa, resta somente o seu processo de fabrico, que será abordado na secção seguinte.

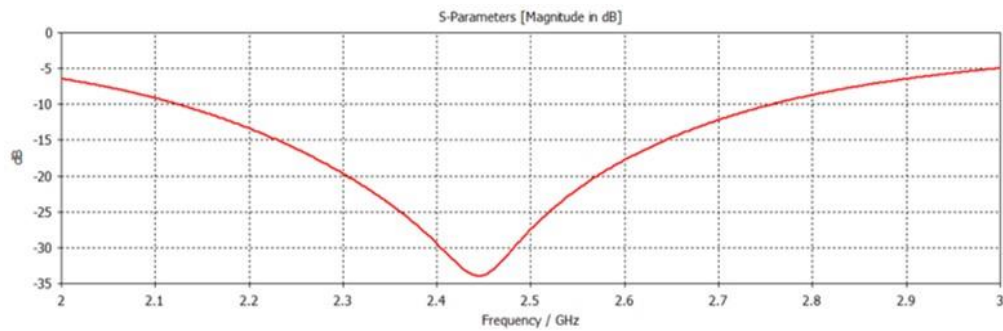


Gráfico 3.5: $|S_{11}|$ em função da frequência

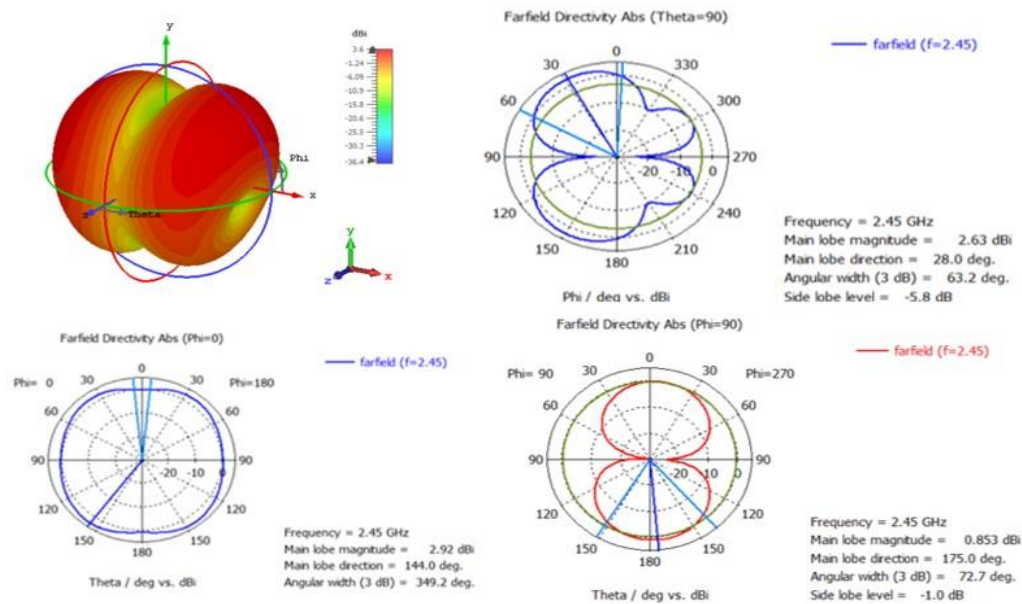


Figura 3.17: Diagramas de radiação do monopolo impresso ($f=2,45$ GHz)

3.4.2 Construção e Teste

O processo fotolitográfico é usado para a produção das antenas impressas, à semelhança do processo de fabrico de placas de circuito impresso. É um processo que utiliza substâncias químicas, de forma a gravar no cobre da placa a geometria desejada para o projeto.

Inicia-se o processo com a limpeza da PCB (placa de substrato coberta de ambos os lados por cobre) com um pano húmido e detergente próprio para remover da superfície gorduras e impurezas. Nesta fase é necessário ter cuidado para não deixar impressões digitais na placa, para garantir boa aderência do produto a ser utilizado no próximo passo.



Figura 3.18: Máquinas utilizadas no processo (decapagem à esquerda e UV à direita)

Uma solução foto-resistente, neste caso *Positiv 20*, é aplicada em toda a superfície de cobre, esperando-se 24 horas que esta seque. Este processo deve ser efetuado numa sala em câmara escura. Esta substância é sensível à luz ultravioleta, o que permite usar uma máscara (figura 3.19) impressa numa película de plástico transparente para delimitar as zonas onde se quer manter a superfície de cobre, criando assim o esquema da antena sobre o substrato.

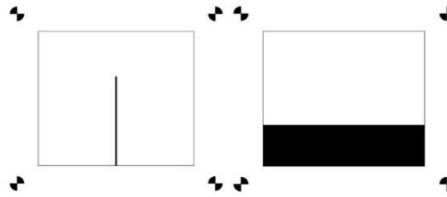


Figura 3.19: Máscara utilizada no processo fotolitográfico

O *Positiv 20* é resistente a uma mistura de soda cáustica e água à exceção dos locais em que foi atingido com radiação UV, permitindo assim obter uma placa com um esquema da antena em cobre sobre o dielétrico. De seguida a substância *Positiv 20* é eliminada e a parte de cobre que não está protegida é sujeita a um processo de decapagem com uma solução de perclorato de ferro. Finalmente o restante da solução foto-resistente é eliminada do cobre com acetona, concluindo o processo de construção da antena impressa com a colocação do conector SMA, soldando-o à linha de alimentação do monopolo.

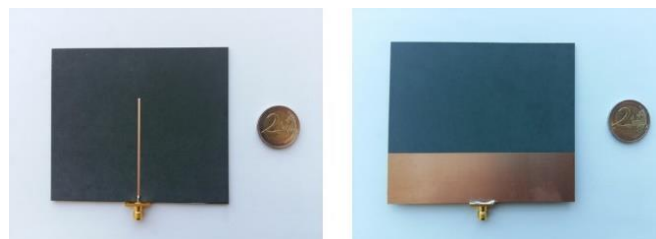


Figura 3.20: Monopolo impresso de 2,45 GHz com conector SMA

Terminada a construção do monopolo impresso, antes da sua aplicação, restava testá-lo. Os testes da antena foram efetuados no Instituto Superior Técnico, tendo-se iniciado pela medição do fator de reflexão. Os resultados medidos não coincidiram com os

resultados simulados, como podemos ver no gráfico abaixo, pois o coeficiente de reflexão mínimo encontra-se nos 2,28 GHz com uma magnitude de -20,5 dBi, o que não seria de esperar.

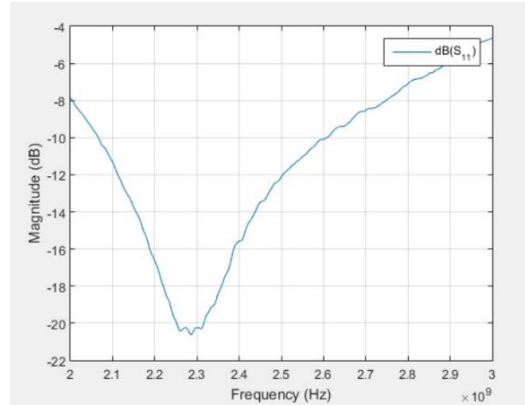


Gráfico 3.6: Primeira versão do coeficiente de reflexão do monopolo impresso

Para contrariar este efeito, que se pode dever à falta de ajuste na malha do *software CST*, foi retirado à altura do monopolo um total de 4 mm, obtendo-se assim a curva $|S_{11}|$ apresentada seguidamente, tendo-se centrado a frequência correspondente ao coeficiente mínimo para os 2,35 GHz.

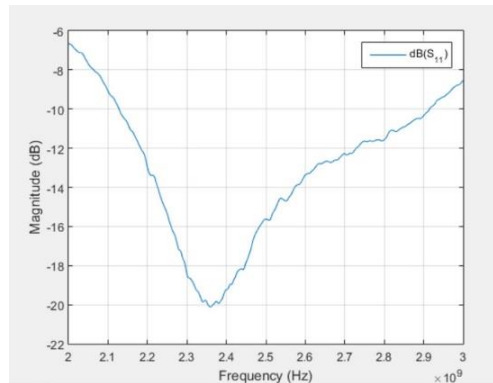


Gráfico 3.7: Versão final do coeficiente de reflexão do monopolo impresso

As medidas do coeficiente de reflexão permitiram-nos concluir que para a frequência central teremos uma potência refletida de aproximadamente 1% e uma largura de banda ($|S_{1,1}| < -10 \text{ dB}$) de 780 MHz.

Concluída a primeira etapa de testes da antena, iniciaram-se os testes na câmara anecoica com o objetivo de fazer a medição do diagrama de radiação. As medições foram efetuadas em dois planos, plano E e plano H, tendo sido medidas as componentes polares e copolar (cruzadas) para ambos os planos.

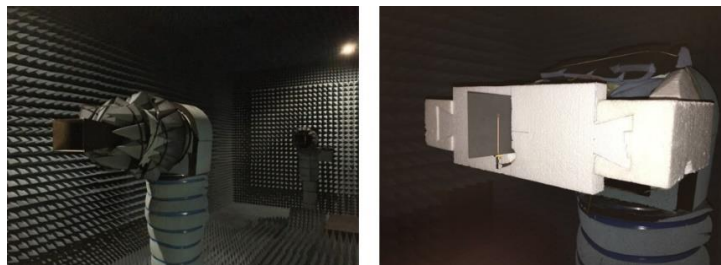


Figura 3.21: Montagem para a medição do diagrama de radiação da antena

Para se obter os diagramas de radiação da antena que se pretende testar, é necessário obter um termo de comparação, que foi atingido com a utilização de uma corneta, pois os seus parâmetros são bem conhecidos. Tendo os valores de base para a medição do diagrama de radiação, colocou-se a antena numa plataforma de esferovite, electromagneticamente invisível, permitindo assim a medição.

Aquando a montagem do circuito para os testes, um dos maiores problemas encontrados em todas as etapas foi o encobrimento de superfícies metálicas ou elementos que pudessem influenciar as medidas, alterando assim a fiabilidade dos resultados.

Após a realização de todos os passos necessários para a obtenção dos valores das medidas, foi necessário processar esses dados, tendo em conta os valores base das medições da corneta. Após algum pós-processamento, representado no gráfico 3.8, podemos concluir que houve interferências nas medições dos parâmetros da antena. Para além da discrepância entre resultados experimentais e simulados em software, podemos também observar na curva da componente cruzada do plano H algumas interferências de componentes de corrente. Estas componentes de corrente são provenientes do cabo coaxial que alimenta a antena, pois este encontrava-se com a mesma orientação em que estavam a ser efetuadas as medidas, interferindo assim com a qualidade dos resultados das medições.

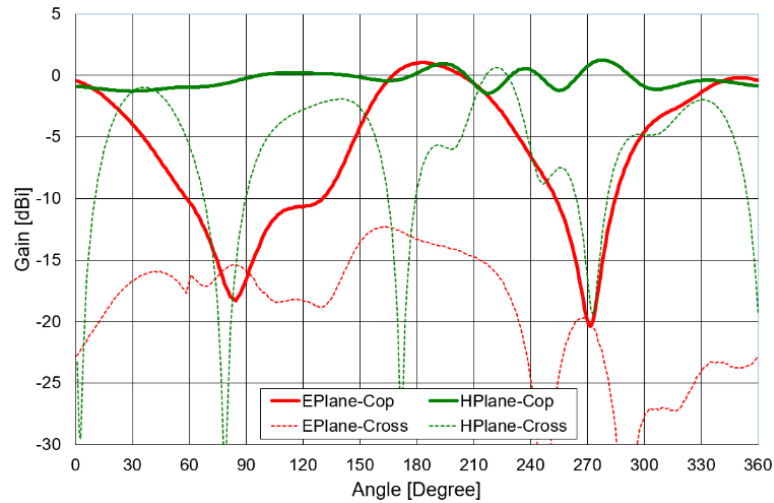


Gráfico 3.8: Curvas do ganho nos diferentes planos das medições

Os erros na medição, impostos pelo cabo de alimentação, influenciaram indiretamente os valores obtidos de ganho, mas não implicaram alterações aos diagramas de radiação esperados, como se pode observar na figura abaixo, em que se compara os diagramas obtidos experimentalmente e os resultados obtidos no simulador *CST*.

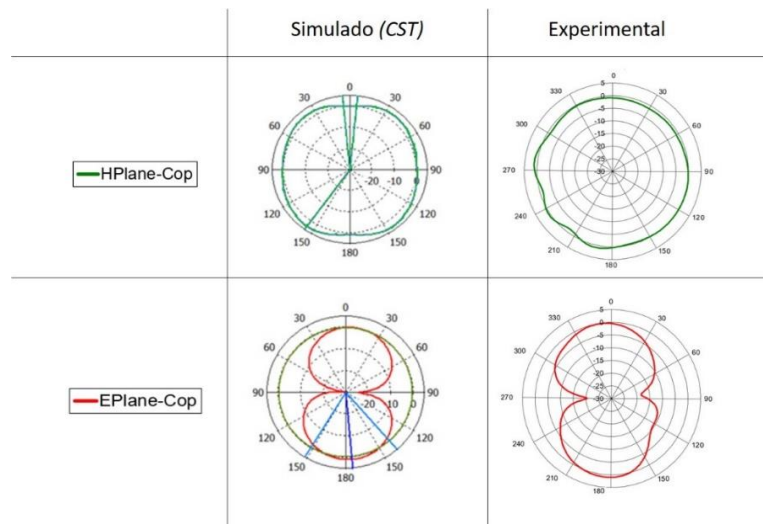


Figura 3.22: Diagramas de Radiação (Simulado vs Experimental)

3.5 Conclusões

A arquitetura do sistema de telecomunicações teve por base a necessidade de responder a qualquer tipo de requisitos que um navio de superfície não tripulado pode apresentar. Planeou-se usar duas bandas de frequência, de forma a obter um canal de telemetria (433 MHz) e um canal de transferência de dados (2,45 GHz).

Neste capítulo procedeu-se ao dimensionamento, construção e teste das antenas necessárias para cada uma das bandas, tendo-se feito um enquadramento teórico prévio.

Para finalizar o sistema foi necessário desenvolver um esquema de linhas de comunicação entre o navio não tripulado e a estação em terra, de forma a fazer a transferência dos dados necessários para o controlo e seguimento por parte do operador. Este esquema encontra-se planificado na figura 3.23.

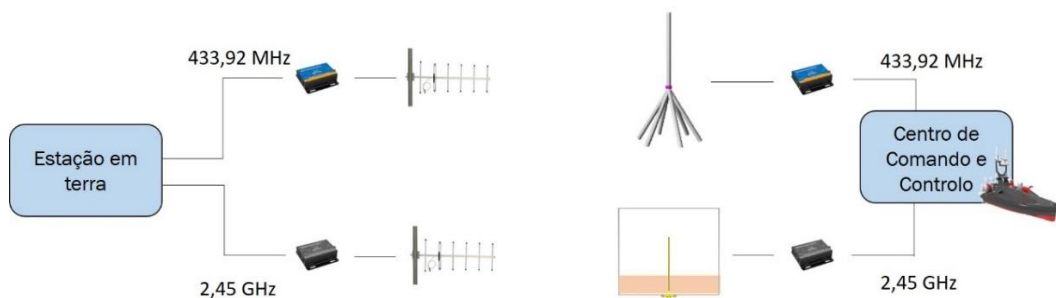


Figura 3.23: Arquitetura do sistema de telecomunicações

A estação em terra será constituída por um equipamento (computador de controlo) que irá agregar todas as informações de telemetria e sensoriais e preparar os dados de controlo a ser enviados para o veículo, sendo este equipamento ligado a um transrecetor, de forma a modular as informações a serem enviadas. Por sua vez, o transrecetor está conectado a duas antenas direcionais, antenas *Yagi-Uda*, uma para cada banda, de forma a aumentar a sua área de ação.

O sistema de telecomunicações do veículo não tripulado será constituído por um computador de bordo que irá fazer a gestão e processamento das informações de todos os sensores que compõem o *USV*, injetando assim essas informações no transmissor de

forma a serem enviadas através das antenas projetadas no presente capítulo. O sentido inverso funcionará de forma semelhante, permitindo assim um fluxo de energia constante.

O desenvolver deste sistema de telecomunicações representa uma vantagem no que toca à liberdade de escolha dos parâmetros que queremos implementar no sistema, mediante os requisitos estabelecidos para o veículo. Se for necessário alterar a banda de frequência ou as características das antenas, podemos seguir o procedimento mencionado durante o capítulo, de forma a obter um canal de comunicação que cumpra as necessidades do projeto.

Durante o processo de análise, projeto e construção das antenas foram encontradas algumas dificuldades relacionadas com o manuseamento dos programas usados para o projeto das antenas, principalmente no *CST*, pois as variáveis disponíveis para simulação dos parâmetros são muitas, acabando por dificultar assim a escolha dos métodos mais indicados.

O trabalho desenvolvido irá, desta forma, permitir a aplicação de duas antenas no veículo, tornando assim possível o fluxo de informação entre os dois elementos necessários à comunicação, garantindo a viabilidade do projeto VENT-SUP-EN.

4 Integração dos Sistemas

O VENT-SUP-EN é um projeto que visa a construção um veículo de superfície, capaz de ser operado remotamente ou que opere de forma autónoma. Essa capacidade exige a existência de certas características e componentes que possibilitem a ausência parcial ou total do Homem.

O controlo remoto de um veículo implica o conhecimento total do seu comportamento e componentes e a possibilidade de o controlar à distância. Para isso, necessitamos de um sistema de telemetria³⁴ aliado a um microcontrolador. Essa interligação permite a aquisição e transmissão de todos os dados sensoriais essenciais à manobra do veículo e desempenho da missão, partilhando a linha de comunicação com a receção dos comandos enviados pelo operador, sendo tudo isto coordenado por um microprocessador que, para além do mencionado, aciona também os atuadores para movimentar a plataforma, ver figura 4.1. A integração a ser estudada neste capítulo irá incidir somente no transreceptor e *datalogger*.

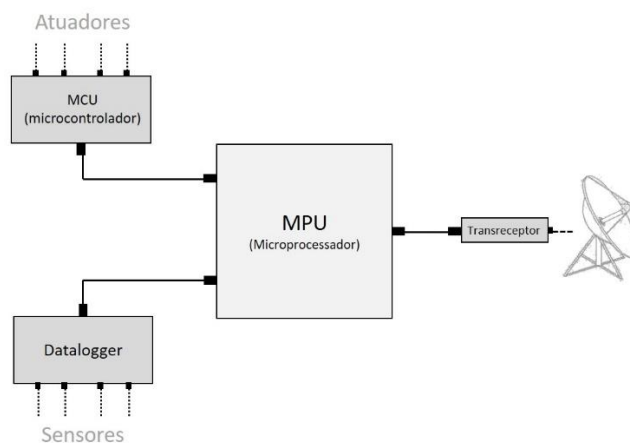


Figura 4.1: Arquitetura do Sistema de Controlo Remoto de um Veículo

³⁴ Telemetria indica como se recolhem dados de uma localização remota e os transmitem de forma eficiente possibilitando assim a sua gravação e análise (Lozano-Nieto, 1999).

Nos capítulos anteriores, foram abordados dois temas que integram a arquitetura de um sistema de controlo remoto, possibilitando telemetria, comando e controlo do veículo. A integração dos dois sistemas (sensores e telecomunicações) será o tema abordado neste capítulo, tendo assim em vista a prova de conceito apresentado anteriormente e demonstração do modo de funcionamento dos dois sistemas.

No presente capítulo irá constar, inicialmente, uma breve descrição da arquitetura concebida para o protótipo do projeto VENT-SUP-EN, indicando as suas capacidades e interfaces de comunicação internas e externas. Compreendida a arquitetura concebida, irá ser detalhado quais os diferentes componentes escolhidos, como se irão interligar e de que forma irão comunicar. Para finalizar o capítulo, explicar-se-á de que forma foram desenvolvidas rotinas para implementar a arquitetura do projeto.

4.1 Conceito

Tendo por base a arquitetura de um sistema de controlo remoto de um veículo, os sistemas previamente desenvolvidos e as necessidades do projeto, foi idealizado um sistema autónomo capaz de:

- Comunicações WiFi (2,4 GHz);
- Comunicações longa distância (433 MHz);
- Aceder aos dados dos sensores;

O sistema idealizado é autónomo, dado que não necessita de um processador externo para gerir os diferentes módulos acoplados, sendo capaz de fazer a aquisição, tratamento e transmissão de dados sensórias via canais de comunicações. De forma a equipar este sistema com facilidades de comunicação e integração com outros sistemas e arquiteturas, foram implementados dois canais para comunicação interna, RS232 e *Ethernet* (RJ45). A figura 4.2 representa o esboço inicial do protótipo para o projeto.



Figura 4.2: Esquema básico do sistema a implementar

No capítulo “Sensores de um Veículo Não Tripulado” foram analisados diversos sensores que estavam organizados por módulos. O primeiro módulo, módulo de navegação, agrupava um conjunto de sensores entendidos como essenciais à navegação, tendo sido analisado o GPS, IMU, RADAR e Sonda. Neste capítulo, estamos interessados em fazer a integração do módulo de navegação com o sistema de telecomunicações estudado no terceiro capítulo. Para efeitos de prova de conceito, irão ser utilizados unicamente dois sensores, GPS e IMU. O RADAR não será utilizado de momento, pois representa um investimento demasiado elevado para a fase do projeto em que nos encontramos. A sonda também não será utilizada, visto que é um sensor que necessita de um meio aquático para produzir resultados e uma vez que ainda não está disponível uma plataforma para a sua instalação, a sua aquisição é desapropriada.

Tendo em mente o esboço do sistema a implementar e os sensores e canais de comunicação que vão ser utilizados, deduzem-se as seguintes atribuições de dados para as interfaces de comunicação:

- RS232:
 1. GPS;
 2. IMU;
 3. Comunicação 433 MHz;
- RJ45:
 1. Comunicação 2,4 GHz;

Os moldes em que foram distribuídos os dados pelas interfaces mencionadas devem-se às seguintes razões: leitura dos sensores e dados *LoRa* (modulação usada na banda de frequência 433 MHz) não requerem uma velocidade de transmissão alta, os 115200 *bps* do *RS232* são suficientes; para dados *WiFi* (protocolo IEEE 802.11 b/g/n usado na banda de frequência 2,4 GHz) é necessário maior taxa de transferência, sendo a interface *Ethernet* mais apropriada.

Concebido o esquema geral da arquitetura do protótipo, é necessário fazer a escolha dos equipamentos que devem constituir a arquitetura. Os sensores GPS e IMU que integrarão o sistema foram discutidos no segundo capítulo, tendo sido escolhidos os seguintes modelos: *U-blox NEO-M8N* (GPS) e *BOSCH BNO055* (IMU). As antenas necessárias para os canais de comunicações foram projetadas, fabricadas e testadas no terceiro capítulo, tendo sido adquirida uma antena para o GPS (*patch* circular - 1,575 GHz). Na próxima secção, irá analisado o microcontrolador e os transreceptores escolhidos para equipar o protótipo.

4.2 Microcontrolador - *ESP32* – *Wemos TTGO t-beam*

ESP32 (Espressif, 2019) é uma família de microcontroladores que estão integrados em diversas placas de desenvolvimento, utilizadas maioritariamente em projetos de *IoT*. O modelo *Wemos TTGO t-beam* foi o modelo de placa de desenvolvimento escolhido para integrar a arquitetura deste projeto. Esta escolha deveu-se, principalmente, ao vasto leque de módulos que vêm integrados na placa: dois módulos transreceptores (2,4 GHz e 433 MHz) com capacidade de conexão a antenas externas, Bluetooth, 10 entradas disponíveis para ligação a sensores, módulo de *GPS*, um sensor de temperatura e o microcontrolador.

Nesta arquitetura, o *ESP32* funciona como plataforma de aquisição e processamento dos dados dos sensores (*GPS* e *IMU*) e módulo de comunicações para as duas bandas que se pretendem implementar, fazendo a transmissão e a receção de dados da estação em terra.

O módulo *Wemos TTGO t-beam* pode ser alimentado a 3,3V ou 5V, consumindo no máximo 250 mA durante transmissões rádio.

O microcontrolador *ESP32* é composto por um processador dual-core de 32 bits (*Xtensa*) com uma velocidade de *CPU* de 240 MHz. Possui uma memória interna *SRAM* de 520kB e uma memória flash externa de 4MB. Com portas multiplexadas, é capaz de permitir: 18 canais *ADC* (*Analog to Digital Converters*), 3 interfaces *SPI*, 3 interfaces *UART*, 2 interfaces *I²C*, 2 interfaces *I²S*, 16 saídas *PWM* (*Pulse Width Modulation*) e 2 canais *DAC* (*Digital to Analog Converters*).

O *ESP32* implementa TCP/IP com protocolo IEEE 802.11 b/g/n, permitindo que o dispositivo seja utilizado como *Access Point* (*softAP*) e *Client* (*station*). A placa de rede comporta uma largura de banda de 20 ou 40 MHz, com uma velocidade máxima de 150 Mbps e com uma potência de transmissão de 20,5 dBm.

Incorporado na placa, encontra-se também um modulo transreceptor (*Semtech SX1276*) (Semtech, 2015) na banda dos 433 MHz com uma potência de transmissão de 20 dBm, consumindo cerca de 170 mA e possibilitando as seguintes modulações: FSK (*Frequency-shift Keying*), GFSK (*Gaussian Frequency-shift Keying*), MSK (*Minimum-shift Keying*), GMSK (*Gaussian Minimum-shift Keying*), OOK (*On-off Keying*) e LoRa (*Long Range*). A modulação utilizada para este projeto foi LoRa. LoRa é uma modulação baseada na tecnologia *chirp Spread spectrum* (*CSS*) introduzida em 2012. Esta tecnologia possibilita comunicações rádio a longas distâncias com um baixo consumo de potência, permite uma largura de banda máxima de 250 kHz e uma taxa de transferência de dados de 37,5 kbps.

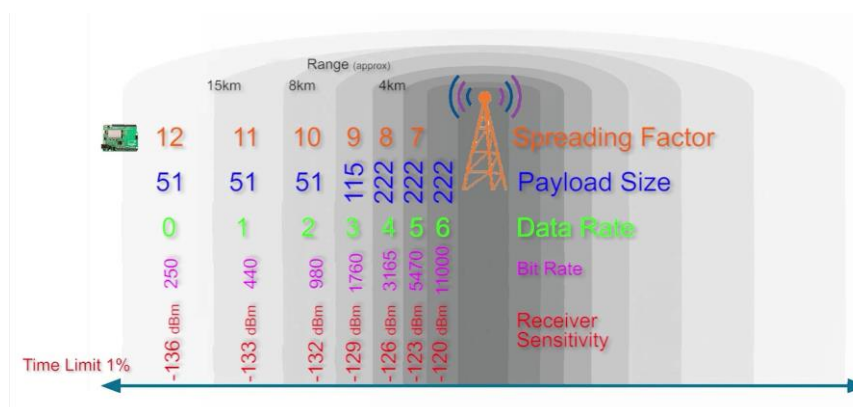


Figura 4.3: Influência dos parâmetros LoRa no alcance

O módulo de *GPS u-blox M8N* está integrado na placa através do protocolo *UART* o que permite uma taxa de transmissão padrão de 9600 bps (*bits per second*). Este módulo é capaz de captar as principais constelações de satélites (GLONASS, BeiDou e Galileo), possui sistema de correção *SBAS*, possibilitando uma precisão da localização de 2 metros num plano horizontal, conseguindo atingir uma taxa de refrescamento da posição de 5 Hz. Em arranque a frio é capaz de fixar uma posição em cerca de 26 segundos, diminuindo este tempo para 1 segundo em arranques quentes.

4.3 Arquitetura do Protótipo

Com todos os módulos que devem equipar o protótipo conhecidos, devemos agora estudar mais detalhadamente como irão ser feitas todas as ligações e integração dos módulos numa arquitetura única. De forma a poder testar as valências e funcionalidade do protótipo, foi necessário adquirir um módulo *Raspberry Pi*, cuja finalidade seria funcionar como ponto de ligação final para as interfaces *RS232* e *RJ45*. O modelo adquirido foi o *Raspberry Pi Model 3B+* (Raspberry Pi, 2018) que possui de origem uma entrada *RJ45*.

A placa *ESP32* apesar da sua polivalência, não possui nenhuma porta *RJ45*, nem *RS232*; o *Raspberry Pi*, por seu lado, não inclui nenhuma ligação *RS232*. Para colmatar estas ausências, foram também adquiridas as facilidades de ligação para estes protocolos. O módulo ***LAN8720 ETH board*** (Microchip, sem data) converte a interface *SPI*, com velocidades relógio de 20 MHz, numa ligação *Ethernet* 10/100Base-T que suporta comunicações *Full* e *Half-Duplex*. O módulo ***RS232 board – WS*** (Waveshare, 2013) converte os níveis *TTL* em níveis *RS232*, com conectores DB9, usando uma interface *UART*. Estes módulos foram implementados de forma a facilitar a integração com qualquer módulo que possa equipar o VENT-SUP-EN em estados mais avançados de desenvolvimento.

A arquitetura detalhada do protótipo pode ser consultada na figura 4.4.

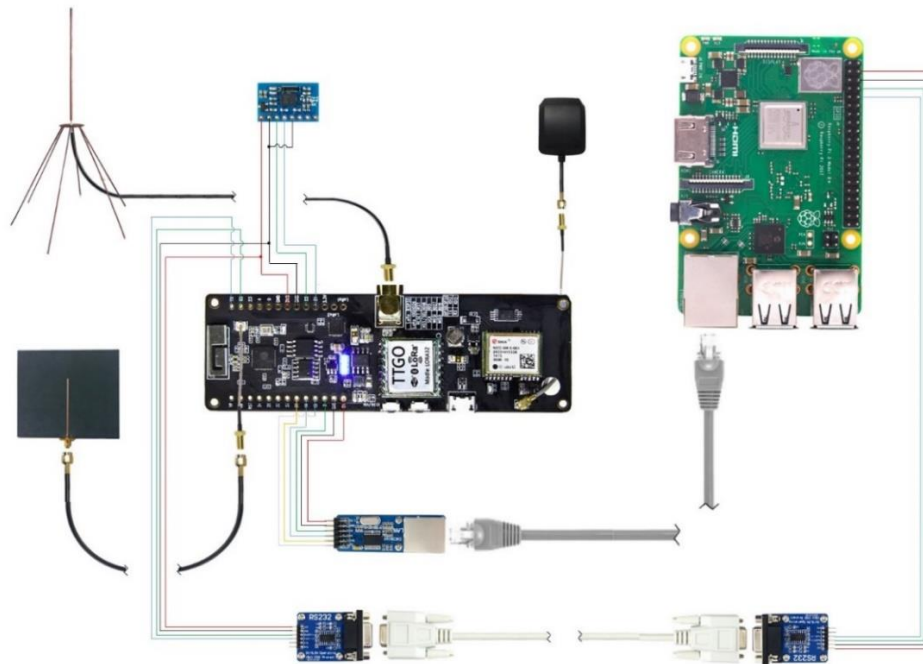


Figura 4.4: Arquitetura do sistema C3³⁵ do VENT-SUP-EN

Com base na arquitetura apresentada e na distribuição de dados para as interfaces de comunicações mencionados anteriormente, é necessária a criação de protocolos de comunicações para fazer a gestão da fluência de dados nas ligações. Na secção seguinte irá ser abordado o protocolo *RS232* criado para a comunicação entre o *ESP32* e o *Raspberry* para troca de dados dos sensores e mensagens LoRa (433 MHz).

4.4 Protocolo RS232

A distribuição de tarefas entre as duas plataformas usadas na arquitetura do protótipo do VENT-SUP-EN implica uma constante comunicação entre estas. De forma a facilitar e a tornar mais robusta a comunicação, foi criado um protocolo a ser implementado nas duas plataformas, permitindo assim a ligação série.

Este protocolo consiste numa conferência com regras bem definidas entre os dois pontos de ligação. Esta conferência é definida pela relação que os dois pontos têm entre

³⁵ Comando, Controlo e Comunicações

si, um é o *master* (*Raspberry Pi*) e o outro é o *slave* (*ESP32*). Quando se trata de dados dos sensores, a conferência é sempre iniciada pelo *master* que envia mensagem de comando (*cmd*), que neste caso são pedidos de informação (dados dos sensores). Após receção da mensagem de comando, o *ESP32* responde com a informação pedida pelo *Raspberry*, denominada por *reply*. Quando se trata de comunicações *LoRa*, a comunicação é bidirecional, tendo o *ESP32* autonomia para enviar a mensagem que recebeu da estação em terra para o *Raspberry*. Quando o *Raspberry* quer enviar uma mensagem via *LoRa*, envia-a ao *ESP32* e este reencaminha-a. Por exemplo, no caso de mensagens periódicas, como mensagens de telemetria em formato NMEA, é o *Raspberry* quem ordena a transmissão da mensagem, ver figura 4.5. Já mensagens recebidas via *LoRa* da estação de terra seriam enviadas para o *Raspberry Pi* por iniciativa do *ESP32*.

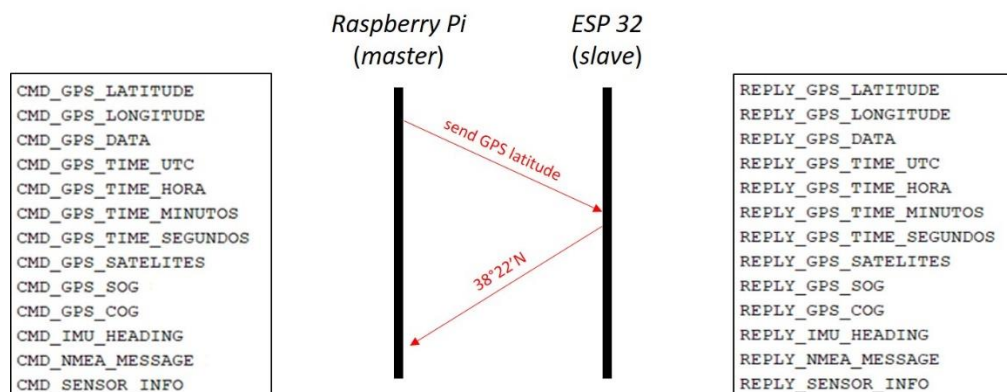


Figura 4.5: Protocolo RS232 e mensagens possíveis

De forma a proporcionar alguma robustez e clareza à comunicação, o protocolo efetua a troca de informações com recurso a mensagens pré-formatadas, onde são introduzidos os dados a enviar de forma encapsulada, ou seja, existe um cabeçalho e um rodapé em cada mensagem que contêm informações sobre os dados a enviar, tipo de mensagem e códigos que previnem a corrupção dos dados e a deteção de erros. A formatação da mensagem correspondente ao protocolo de série criado para esta comunicação pode ser facilmente compreendida através da análise da figura 4.6. O código que implementa este protocolo encontra-se no apêndice A.

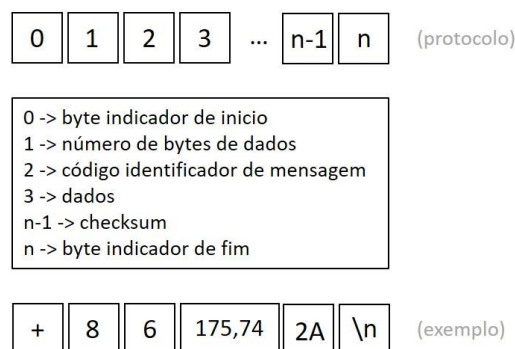


Figura 4.6: Mensagem formatada do protocolo *RS232*

4.5 Ligação *Ethernet*

À semelhança da comunicação série, era também objetivo inicial para esta arquitetura providenciar uma solução com uma interface RJ45. Tal não foi possível por duas razões: A primeira prendeu-se com a dificuldade encontrada quando se tentou conjugar as rotinas dos dois protocolos (*Ethernet* e *RS232*) na plataforma *ESP32*, ocorrendo uma sobrecarga desta plataforma, sendo difícil manter o sincronismo e gerir tantas fontes de informação simultaneamente. A segunda prendeu-se com a falta de experiência em linguagem *Python*, usada na programação do *Raspberry*, tendo sido unicamente aprendida no decorrer do projeto.

Em alternativa à arquitetura apresentada, foi também idealizada uma solução para colmatar a incapacidade de utilizar o transrecetor *WiFi* do módulo *ESP32*, que passa pela utilização de uma placa de rede diretamente ligada à antena de 2,4 GHz que, por sua vez, se conectará à porta RJ45 do *Raspberry*. Na figura 4.7 é utilizado um router da TP-Link em *station mode*, criando assim uma ponte *wireless* entre a estação em terra e o veículo.

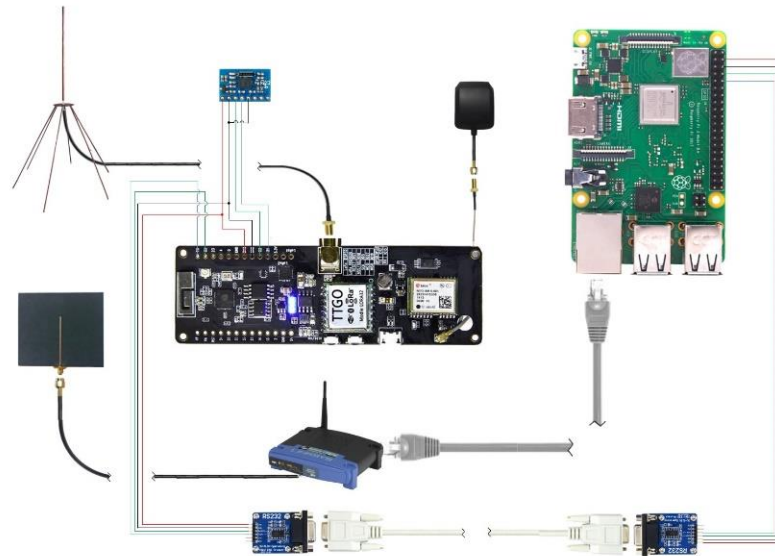


Figura 4.7: Arquitetura alternativa do protótipo VENT-SUP-EN

4.6 Programação: Rotinas e Tarefas

A programação do microprocessador da placa *Raspberry Pi* foi possível após a instalação do sistema operativo *Raspbian* e do programa de compilação que executa a linguagem *Python*. Estando todas as facilidades de introdução de *firmware* no módulo terminadas, passou-se para a configuração das portas série, através do *LX Terminal*, para permitir a utilização do protocolo *UART* e permitir a ligação por *RS232* ao *ESP32*. Com vista a implementação do protocolo acima referido para a comunicação série, foi delineada uma estratégia para a elaboração do *software* que permitisse a transmissão dos códigos de mensagem e a receção dos dados, através da ligação série, e executasse uma rotina para detecção de gralhas nas mensagens.

A elaboração do *software* a ser executado no *RPi* foi organizada em 5 secções: a primeira secção consistiu na importação de todas as bibliotecas; a segunda secção consistiu na configuração da porta série; a terceira será a função onde se faz o pedido, leitura e manipulação dos dados vindos do *ESP32*; a quarta secção apresenta uma função que tem a preocupação de verificar se os dados recebidos estão corrompidos, fazendo uma comparação entre o *checksum* interno e externo; e finalmente, na última secção, a

rotina é executada ciclicamente, fazendo a chamada das diferentes funções. No apêndice C é apresentada a programação descrita.

Para programar o microcontrolador *ESP32* recorreu-se ao ambiente de desenvolvimento do *Arduíno*, tendo sido necessário obter atualizações de *software* a partir do gestor de placas e renovar as bibliotecas.

O desenvolvimento do programa do microcontrolador *ESP32* foi idealizado de forma a implementar o protocolo de comunicação série apresentado anteriormente, fazer a aquisição dos dados dos sensores e fazer a gestão dos módulos de comunicações. Depois de terem sido carregadas as bibliotecas, declaradas as variáveis e feito as configurações das portas a serem utilizadas foram criadas as seguintes funções: aquisição do valor da proa (*IMU*), aquisição de todos os dados do *GPS* (latitude, longitude, registo horário, número de satélites, etc), leitura de códigos do protocolo série vindos do *RPi* e construção de todos os componentes do protocolo (*checksum*, byte de início, byte de fim, código e tamanho de mensagem). No apêndice B é apresentada a programação descrita.

4.7 Conclusões

O presente capítulo teve como objetivo uma prova de conceito, que consistiu na integração dos sistemas analisados e desenvolvidos durante a dissertação: sistemas de sensores e sistema de telecomunicações. Inicialmente, foi feita uma breve descrição da arquitetura do protótipo, indicando as suas capacidades e interfaces de comunicação internas e externas. Foram enunciados os componentes escolhidos, como se interligaram e de que forma comunicam. Terminou-se o capítulo com a explicação das rotinas implementadas.

Relativamente ao *software* desenvolvido, a robustez do protocolo série apresentado para a placa *Raspberry* ficou por completar, sendo que só executava o pedido dos dados sensoriais e transmissão de comandos para os módulos de comunicações, através dos códigos identificadores de mensagem, não havendo um correto encapsulamento da mensagem, deixando-a assim suscetível a corrupção. A robustez dos códigos poderia também ser melhorada acrescentando mais rotinas de deteção de erros, pois as que se

encontram no código são unicamente acionadas se algum dos módulos não inicializar ou se, por algum motivo, a função esperada não for desempenhada.

Na seguinte tabela estão indicados todos os componentes utilizados na arquitetura implementada neste projeto, com os respectivos preços de aquisição. Conclui-se que a integração dos sistemas desenvolvidos e criação de um protótipo para o projeto não é um processo que envolva custos demasiado elevados. Não foram contemplados os custos de desenvolvimento e material das antenas, sendo que a tabela unicamente representa as despesas necessárias para a aquisição de equipamentos.

Equipamento		Preço
Antena de fio	Conector SMA	7,30€
	Cabo SMA	7,11€
	Adaptador U.FL - SMA	1,60€
Antena impressa	Conector SMA	8,06€
	Cabo SMA	2,29€
	Adaptador U.FL - SMA	1,60€
Antena <i>GPS</i>	Antena	3,90€
	Adaptador U.FL - SMA	1,60€
Sensor <i>IMU</i>	<i>BOSCH BNO055</i>	12,59€
Computador	<i>Raspberry Pi Model 3B+</i>	32,44€
Microcontrolador	<i>ESP32 Wemos TTGO t-beam</i>	25,25€
Ligações	<i>LAN7820 ETH board</i>	5,61€
	<i>RS232 board – WS (×2)</i>	6,34€
	<i>Cabo RS232 (DB9)</i>	3,30€
	<i>Cabo Ethernet (RJ45)</i>	2,80€
	<i>Jumper Wires</i>	3,85€
		125,64€

Tabela 4.1: Material utilizado no protótipo do VENT-SUP-EN

5 Testes e Análise de Resultados

Tendo como principal objetivo a avaliação do desempenho do protótipo apresentado, foram realizados testes ao sistema de telecomunicações, pondo assim à prova todo o sistema de telemetria para os dois canais de transferência de dados.

A avaliação de desempenho foi possível através da criação de um circuito de testes que consistia em duas estações: a primeira, a estação em terra, equipada com uma antena direcional, acoplada a uma placa de rede capaz de correr uma série de testes para avaliar o desempenho de diferentes parâmetros em função da distância a que se encontra da segunda estação; e a estação móvel, essa segunda estação, que estaria equipada com uma plataforma constituída pelo sistema de telemetria, figura 5.1.



Figura 5.1: Plataforma móvel de testes

5.1 Banda 433 MHz

Os testes para a banda de frequência de 433 MHz decorreram na Base Naval de Lisboa e no Rio Tejo, estando a estação em terra localizada no topo do edifício escolar da Escola Naval (antena a 28 metros de altura) e a estação móvel, que se encontrava montada a bordo da embarcação da classe Mindelo do CNOCA (antena a 1 metro de altura). Esta estratégia ajudou a diminuir ao máximo as interferências de edifícios na linha de vista e na zona de *Fresnel*³⁶. O teste de alcance foi efetuado no Rio Tejo, em mar com ondulação ligeira e céu limpo. Em alguns momentos, existiu a interferência de vegetação e navios de passageiros na zona de *Fresnel*, figura 5.2.

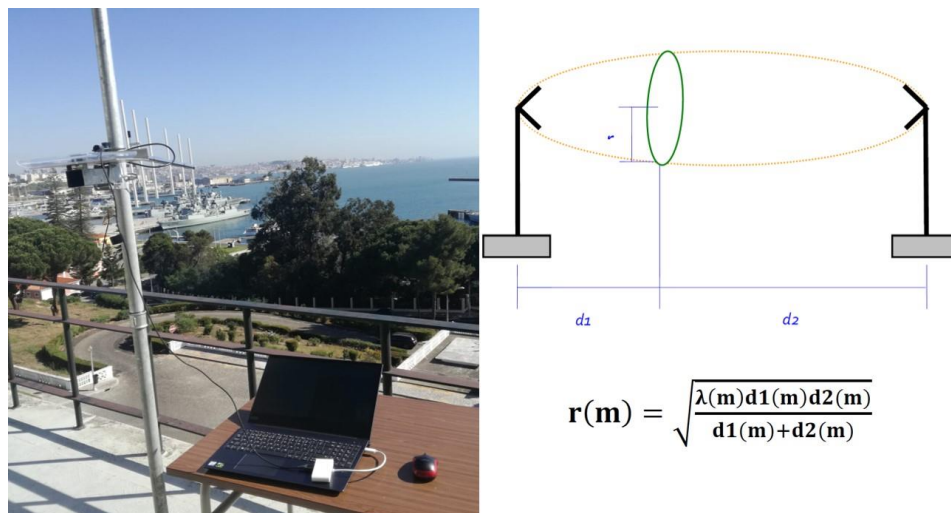


Figura 5.2: Estação em Terra e Zona de *Fresnel*

5.1.1 Arquitetura da Plataforma de Testes

A plataforma de testes foi preparada de forma a cumprir com os objetivos dos testes para esta banda, que passaram pela receção dos dados sensoriais na estação em terra, de forma a observar os movimentos da embarcação ao longo do seu percurso no software

³⁶ A Zona de *Fresnel* define um lugar geométrico tridimensional de forma elipsóide, compreendido entre os dois pontos de comunicação (Seybold, 2005). Se se garantir que não há obstruções nesta zona, não existirão quaisquer tipos de interferências devido a multi-percursos das ondas electromagnéticas. Desde que seja salvaguardada 60% da Zona de *Fresnel*, não se encontram perdas significativas de energia

*OpenCPN*³⁷ e observar a variação da relação sinal ruído em função da distância. A arquitetura da estação em terra e da estação móvel pode ser observada na figura 5.3.

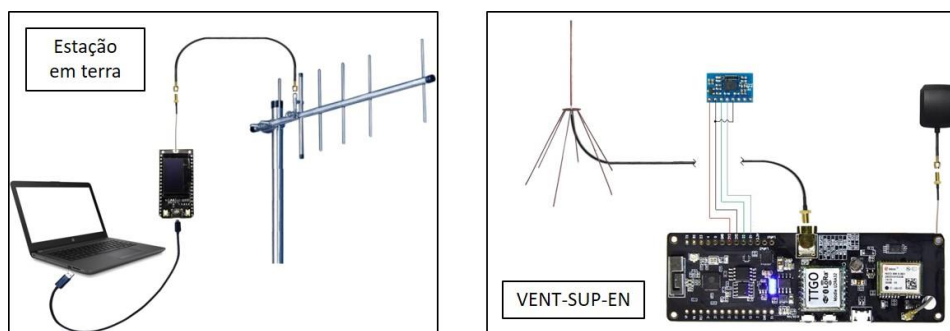


Figura 5.3: Arquitetura da plataforma de testes (banda 433 MHz)

Tendo em conta a arquitetura das estações e os objetivos definidos, foi elaborado *firmware* para tornar os testes passíveis de serem realizados. O *firmware* carregado na estação em terra, apêndice D e E, tinha como principal função a requisição e receção dos pacotes de dados enviados pela estação móvel, disponibilizando-os para a plataforma *OpenCPN* e fazendo uma análise da amplitude do sinal para cada pacote recebido. O *firmware* carregado na estação móvel, apêndice D e F, permitia a aquisição e manipulação dos dados dos sensores e a sua transmissão, após requisição por parte da estação em terra, em formato NMEA através da modulação *LoRa* pelo *chip SX1276* presente no *ESP32*, com seguintes os parâmetros: 20 dBm de potência de transmissão; 7 como *spreading factor*; 125 kHz de largura de banda; taxa de codificação de 4/5; e um preâmbulo com 8 símbolos. A seguinte tabela apresenta o material adicional necessário para completar esta arquitetura para além do material já utilizado na arquitetura do protótipo.

Equipamento		Preço
Estação em terra	<i>ESP32 DevKit</i>	24,96€
	Antena Yagi Laird YS4306	87,06€
	Cabo SMA	7,11€
	Adaptador tipo N - SMA	7,60€
		126,73€

Tabela 5.1: Material utilizado na plataforma de testes (banda 433MHz)

³⁷ *OpenCPN* é uma ferramenta livre de representação gráfica de navegação. A ferramenta é capaz de receber sinais de sensores em formato *NMEA* e representá-los graficamente sobre cartas náuticas eletrónicas

5.1.2 Análise de Resultados

Considerando as condições em que se realizaram os testes, a arquitetura da plataforma descrita anteriormente permitiu que os testes fossem executados até não serem recebidos quaisquer pacotes na estação em terra, tendo obtido uma distância de comunicação de 5738 metros (aproximadamente 3 milhas náuticas) quando a embarcação se deslocou em linha de vista (em direção à ponte Vasco da Gama). Foram também efetuados testes de forma a introduzir interferência na zona de *Fresnel* (em direção à ponte 25 de Abril), tendo-se obtido uma distância de 2670 metros, tendo ocorrido falhas na receção de alguns pacotes na passagem junto a Cacilhas.

Todo o percurso da embarcação foi gravado na ferramenta *OpenCPN* e observados no monitor série os dados recebidos na estação em terra, figura 5.4.

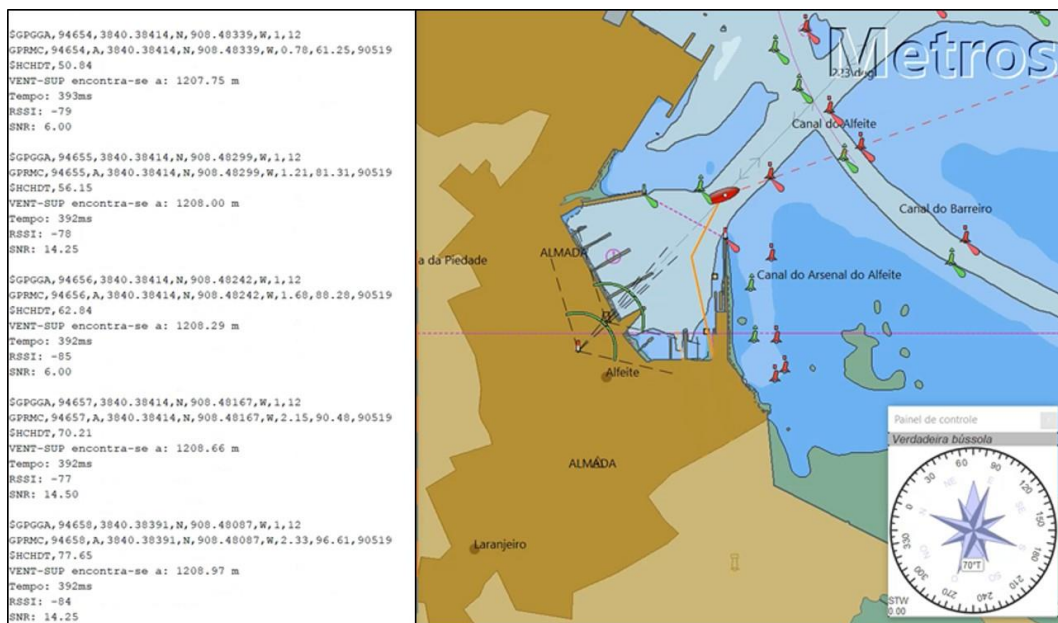


Figura 5.4: Dados de telemetria observados na estação em terra

Os testes efetuados permitiram atingir os objetivos definidos para o sistema de telemetria, atingindo a distância referida.

Na *Fundan University* em Xangai (Li, L., Ren, J., & Zhu, Q., 2017) foram levados a cabo testes semelhantes aos realizados. Os testes foram realizados entre duas estações,

uma em terra com 20 metros de altura e uma estação móvel desempenhada por uma embarcação. A distância máxima obtida foi cerca de 2300 metros, com os seguintes parâmetros: antena com 6 dBi de ganho, banda de 433 MHz, 9 no *spreading factor* e 250 kHz de largura de banda.

Comparando os resultados obtidos nos testes descritos anteriormente com os testes efetuados durante esta dissertação podemos concluir que, tendo em conta algumas diferenças entre parâmetros, os resultados obtidos foram bastante positivos, o que revelam uma montagem de circuito satisfatória, sendo que se se utilizassem amplificadores de sinal e se se otimizassem os parâmetros do SX1276 será expectável obter um aumento considerável no alcance.

5.2 Banda 2,4 GHz

Os testes para a banda de frequência de 2,4 GHz decorreram na bacia de manobra da Base Naval de Lisboa, estando a estação em terra localizada na varanda do edifício do CNOCA (antena a 5 metros de altura) e a estação móvel era uma plataforma a bordo da embarcação da classe Mindelo do CNOCA (antena a 1 metro de altura). O teste de alcance foi efetuado na bacia de manobra, em mar chão, céu nublado e com períodos de aguaceiros.

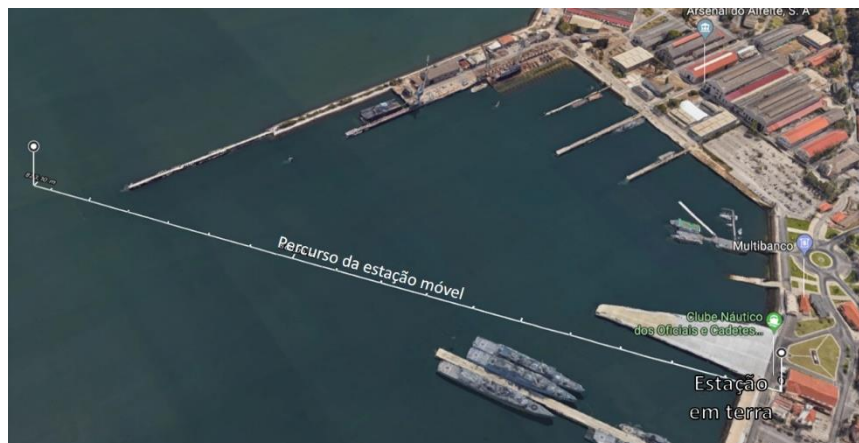


Figura 5.5: Localização da estação em terra e percurso da embarcação

5.2.1 Arquitetura da Plataforma de Testes

A plataforma de testes foi preparada de forma a cumprir com os objetivos dos testes para esta banda, que passaram pela análise dos seguintes parâmetros em função da distância: relação sinal ruído (SNR), nível do sinal recebido, os tempos de latência e a taxa de transferência. Os testes foram executados a diferentes distâncias, sendo que a embarcação fez movimentações de 100 metros, e a cada paragem foram feitas as medições dos diferentes parâmetros.

A análise dos três primeiros parâmetros indicados anteriormente foi conseguida através da utilização de um router na estação em terra funcionando como ponto de acesso e na estação móvel um *ESP32* configurado como servidor. A tabela 5.2 apresenta restante material necessário para construir a plataforma de testes e o respetivo preço.

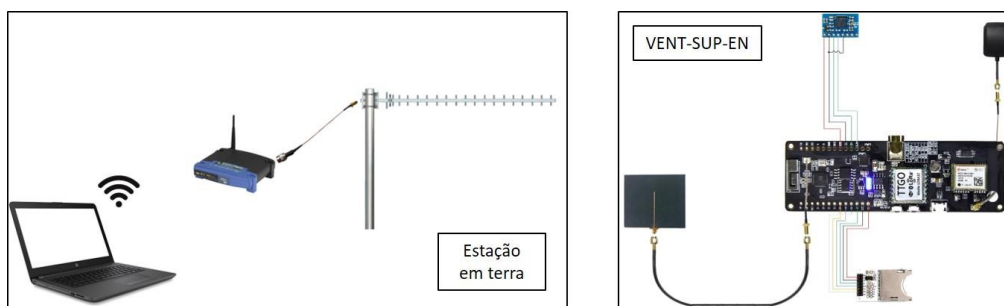


Figura 5.6: Arquitetura da plataforma de testes (banda 2,4 GHz)

Equipamento		Preço
Estação em terra	<i>Router Linksys WRT54G</i>	34,46€
	Antena Yagi 2,4 GHz 26 dBi	15,99€
	Adaptador TNC-SMA	4,30€
	Adaptador RPSMA-SMA	2,40€
VENT-SUP-EN	<i>SD card reader Velleman VMA304 (Velleman, 2017)</i>	4,63€
		57,15€

Tabela 5.2: Material utilizado na plataforma de testes (banda 2.4 GHz)

A obtenção da taxa de transferência utilizou o princípio descrito anteriormente em conjunto com a utilização de um leitor de cartões *SD*, que continha os ficheiros para

download. O *firmware* utilizado no *ESP32* está no apêndice G. O código descreve como foi criada uma página *web* que disponibilizava os ficheiros presentes no cartão para ser executado o download. A figura 5.7 apresenta a página referida.



Figura 5.7: Página *Web* de gestão de ficheiros na estação em terra

5.2.2 Análise de Resultados

O *software* que equipava o router utilizado revelou-se limitado, não permitindo a alteração e medição de certos parâmetros essenciais para a realização dos testes. Após atualização do *software* (*DD-WRT*), foi possível alterar a potência de transmissão e analisar a relação sinal ruído (*SNR*), verificar a capacidade de transmissão do canal e observar graficamente a taxa de transmissão de dados.

A taxa de transferência de dados foi obtida através do download de um ficheiro de 100 megabytes, presente no cartão *SD*, obtido através da página *web*. Estas transferências foram praticadas, tentativamente, a cada 100 metros. Foi registado o valor mais alto durante a transferência do ficheiro.

A relação sinal ruído (*SNR*, *Signal to Noise Ratio*) foi obtida através do *software* do router que analisa este parâmetro para qualquer dispositivo que a este esteja conectado. Este parâmetro indica a diferença entre a intensidade do sinal e o ruído (*noise floor*), sendo possível estimar através deste valor se a comunicação que tentamos estabelecer é viável ou não. Quanto mais alto o valor, melhor desempenho iremos obter na transferência

dados e ligação ao dispositivo em rede. A tendência apresentada no gráfico 5.1 é a esperada, pois a SNR diminui com o aumento da distância.

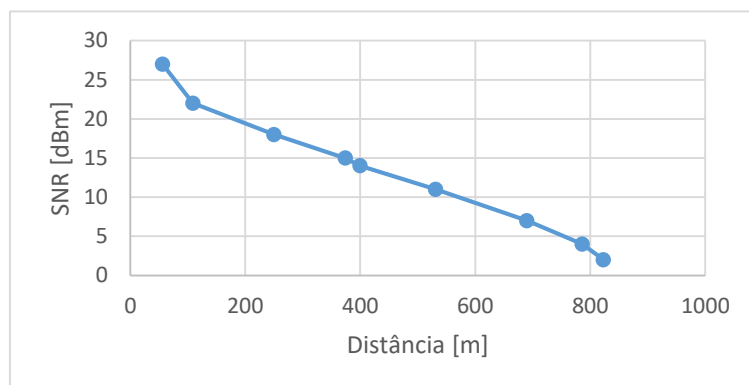


Gráfico 5.1: Relação sinal-ruído no router

A intensidade do sinal é a potência, em dBm, do sinal recebido na antena recetora. Quanto maior o valor, melhor a relação sinal ruído, que por sua vez melhora o desempenho da ligação estabelecida. A tendência apresentada no gráfico 5.2 é a esperada, podendo ser concluído que o *noise floor* é de -90 dBm e a partir deste valor podemos dar como terminada a nossa ligação.

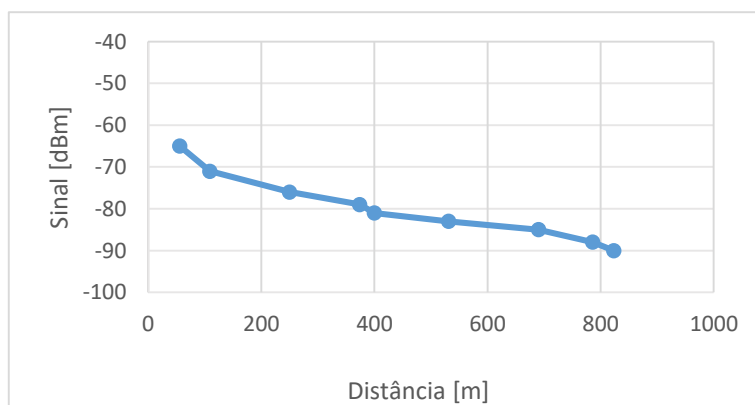


Gráfico 5.2: Nível de sinal recebido

A análise da taxa de transferência de dados permite ao utilizador retirar várias conclusões. Tendo em conta que para fazer o *streaming* de um vídeo com baixa qualidade e baixa taxa de refrescamento (704×480@1fps) é necessário ter uma taxa de transferência

de 48 kbps, pode concluir-se que a distância máxima obtida com *link* de vídeo foi de 750 metros.

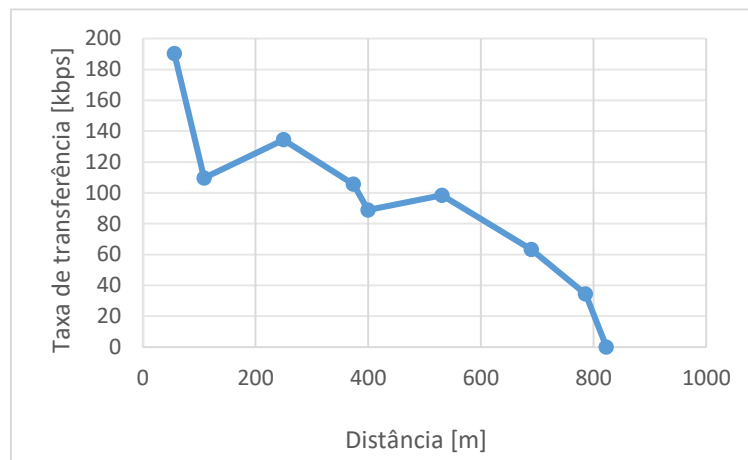


Gráfico 5.3: Taxa de transferência (*download*)

Os últimos parâmetros observados foram os tempos de latência máximo, médio e mínimo, sendo estes tempos obtidos através de uma aplicação (*Colasoft Ping Tool*³⁸). Enviaram-se, em cada distância de teste, 50 pacotes com 32 Bytes de dados, analisando o tempo desde que este foi enviado até haver uma resposta, estando vertido no gráfico 5.4 os tempos obtidos ao longo da distância.

³⁸ Download disponível em https://www.colasoft.com/download/products/download_ping_tool.php

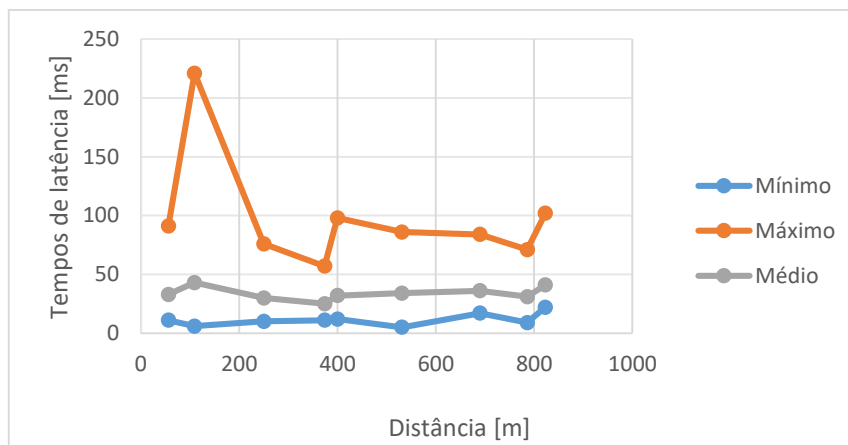


Gráfico 5.4: Tempos de latência

Terminados todos os testes, analisando pormenorizadamente os resultados obtidos e tendo como comparação outras tecnologias existentes, conclui-se que os resultados foram satisfatórios pois conseguiu-se transmitir vídeo a uma distância de cerca de 750m, apesar de ser de baixa qualidade e com apenas 1 *fps*. Seria também benéfico realizar os mesmos testes, mas com uma placa de rede externa, como sugerido no capítulo quatro, para ser possível fazer comparações e tirar conclusões mais aprofundadas.

Na Escola Naval em 2008 (Capela, G. G., Santos, N. P., Vieira, J. M., & Monsanto, C. B., sem data) foram levados a cabo testes semelhantes aos realizados. Os testes foram realizados entre duas estações, uma em terra com 28 metros de altura e uma estação móvel desempenhada por uma embarcação. A distância máxima obtida foi cerca de 7490 metros, com os seguintes parâmetros: antena com 10 dBi de ganho, banda de 2,4 GHz e amplificadores de sinal de 2 *Watts*.

Comparando os resultados obtidos nos testes descritos anteriormente com os testes efetuados durante esta dissertação podemos concluir que, tendo em conta algumas diferenças entre parâmetros, os resultados obtidos foram bastante positivos, visto que a principal diferença se prende com a diferença substancial no ganho das antenas e na potência de transmissão, o que permitem o aumento de distância e qualidade de sinal recebido.

6 Conclusão

A presente dissertação surgiu no seio do projeto detalhado da construção de um Veículo Não Tripulado de Superfície (VENT-SUP-EN), tendo como principais objetivos: o estudo inicial dos sensores que podem equipar o veículo; construção e operacionalização do sistema de sensores básico para navegação; projeto, fabrico e teste das antenas do sistema de telecomunicações do veículo; e a integração dos dois sistemas. Tendo em conta os objetivos definidos foi traçado um plano para os cumprir, organizando a dissertação para ir ao encontro desses objetivos.

Primeiramente, desenvolveu-se um enquadramento teórico dos aspetos que envolvem este estudo. Foram feitas anotações históricas acerca da origem dos veículos não tripulados e das circunstâncias em que foram inicialmente utilizados; do que motiva o estudo e desenvolvimento dos mesmos; de quais as missões que devem desempenhar; do âmbito em que consiste o projeto VENT-SUP-EN e quais as contribuições que esta dissertação pretende efetuar para o projeto.

O estudo do sistema de sensores iniciou-se com uma análise das necessidades sensoriais para o projeto, recorrendo aos pré-requisitos estabelecidos numa publicação referente a veículos não tripulados, tendo estes sido organizados em módulos de sensores. Considerando os veículos já existentes no mercado, foi feito um levantamento de quais os sensores que deveriam equipar o veículo, concluindo-se que as escolhas feitas permitiriam desempenhar 12 de 17 tarefas previstas para o veículo, ficando ainda por analisar os restantes módulos de sensores para atingir os 100% de desempenho de tarefas.

O estudo do sistema de telecomunicações procurou dar resposta a qualquer tipo de exigência a nível de comunicações, tendo sido escolhidas duas bandas de frequência: uma de telemetria para comunicações a longa distância (*LoRa* – 433 MHz) e outra com grande largura de banda (*WiFi* – 2,4 GHz). Com base na escolha das bandas, foi desenvolvido um estudo teórico-prático para cada uma: projeto, fabrico e teste das antenas, em que se concluiu que todos os ensinamentos adquiridos e apresentados neste estudo podem ser

aplicados para desenvolver um sistema de telecomunicações para diferentes projetos, mesmo utilizando outras bandas de frequência.

A prova de conceito consistiu na integração dos sistemas mencionados anteriormente: sistemas de sensores e sistema de telecomunicações. Foi desenvolvida a arquitetura do protótipo a ser construído, a descrição das funções dos diferentes componentes e respetivos *firmwares* para as duas placas de desenvolvimento presentes na arquitetura, que permitiam o desempenho das funções previstas para este protótipo. Com os resultados obtidos, é possível concluir que, apesar das falhas em robustez, os *firmwares* desenvolvidos é um excelente ponto de partida para a construção do centro de comando e controlo do veículo, apresentando soluções para todas as necessidades que possam surgir.

Os testes de campo efetuados incidiram na transferência de dados entre a estação em terra e o veículo. Foi efetuado um teste para cada banda de frequência, analisando o desempenho do sistema de telecomunicações e provando também o conceito de telemetria. Para a banda de telemetria (433 MHz) foi possível atingir cerca de 6 quilómetros, com recurso à transferência de dados sensoriais, para observar o movimento do veículo no seu percurso, ficando por analisar qual o comportamento do sistema com a alteração dos parâmetros do módulo *LoRa* e com a introdução de amplificadores para aumento na amplitude do sinal. Para a banda dos 2,4 GHz, foram considerados diferentes parâmetros, posteriormente analisados quanto ao seu desempenho em função da distância.

Os testes foram satisfatórios pois, tendo em conta que as antenas a bordo não são direcionais, conseguiu-se comunicar com uma estação de terra situada a cerca de 6 km e se conseguiu transmitir vídeo a uma distância de cerca de 750m, apesar de ser de baixa qualidade e com apenas 1 fps.

No futuro, considera-se pertinente desenvolver trabalho no sentido de:

- Introdução de amplificadores de sinal e realização de testes para averiguar o aumento da distância máxima a que é possível realizar comunicações nessa nova situação;

- Automação do veículo, construção do sistema de controlo tendo em conta os dados sensoriais e as missões a desempenhar – movimentação em *zigzag* numa *box* evitando obstáculos, por exemplo.

Em virtude de todos os testes efetuados e trabalho desenvolvido durante esta dissertação, pode concluir-se que os objetivos propostos inicialmente foram atingidos de forma satisfatória.

Referências Bibliográficas

- Ahmad, N., Ghazilla, R. A. R., Khairi, N. M., & Kasi, V. (2013). Reviews on Various Inertial Measurement Unit (IMU) Sensor Applications. *International Journal of Signal Processing Systems*, pp. 256–262. Obtido de <https://pdfs.semanticscholar.org/4e18/9da5f9798a2a1caabd3dfb96ff794768a625.pdf>, acessado em outubro de 2018
- Almeida, C., Franco, T., Ferreira, H., Martins, A., Santos, R., Almeida, J. M., ... Silva, E. (2009). Radar Based Collision detection developments on USV ROAZ II. *OCEANS '09 IEEE Bremen: Balancing Technology with Future Needs*, pp. 1–6. Obtido de http://users.isr.ist.utl.pt/~jcarvalho/Publications_files/Radar_based_collision_detection_developments_on_USV_ROAZ_II.pdf, acessado em abril de 2018
- ASV. (2017). C-Worker 5 [Fact sheet]. Obtido de <https://geomatching.com/uploads/default/m/i/migrationnllxzy.pdf>, , acessado em abril de 2018
- Autoridade Nacional de Comunicações. (2010). Quadro Nacional de atribuição de frequências. Obtido de https://www.anacom.pt/streaming/qnaf20092010_07042010.pdf?contentId=1019281&field=ATTACHED_FILE, acessado em setembro de 2018
- Axis. (2017). Q8741-E Bispectral PTZ Network Camera [Fact sheet]. Obtido de https://www.axis.com/files/datasheet/ds_q8741e_t10093113_pt_1905.pdf, acessado em setembro de 2018
- Balanis, C. A. (2005). Linear Wire Antennas. Em *Antenna Theory* (pp. 151–214). A John Wiley & Sons, INC.
- Barrou, O., El Amri, A., & Reha, A. (2017). Comparison of Feeding Modes for a Rectangular Microstrip Patch Antenna for 2.45 GHz Applications. *Advances in Ubiquitous Networking* 2, pp. 457–469. Obtido de https://www.researchgate.net/publication/309686274_Comparison_of_Feeding_Modes_for_a_Rectangular_Microstrip_Patch_Antenna_for_245_GHz_Applications/link/5c2a2f6ba6fdccfc70741ee7/download, acessado em dezembro de 2018
- Belbachir, A. N. (2010). Imaging Technologies and Architecture of Smart Cameras. Em *Smart Cameras* (pp. 245–257). Springer.
- Bertram, V. (2008). Unmanned Surface Vehicles – A Survey. *Skibsteknisk Selskab, Copenhagen, Denmark*, pp. 1–14. Obtido de http://www.skibsteknisk.dk/public/dokumenter/Skibsteknisk/Downloadmateriale/2008/10_marts_08/USVsurvey_DTU.pdf, acessado em maio de 2018
- Bosch. (2014). BNO055 Intelligent 9-axis absolute orientation sensor [Fact sheet]. Obtido de https://cdn-shop.adafruit.com/datasheets/BST_BNO055_DS000_12.pdf, acessado em novembro de 2018
- Capela, G. G., Santos, N. P., Vieira, J. M., & Monsanto, C. B. (sem data). Telemetria utilizando a norma IEEE 802.11g, pp. 1–12, acessado em Setembro de 2019
- Carter, G. C., Mehta, S. K., & McTaggart, B. E. (2009). Sonar systems. Em *Advanced*

- Signal Processing: Theory and Implementation for Sonar, Radar, and Non-Invasive Medical Diagnostic Systems* (pp. 381–406). CRC Press.
- Center, N. O. and A. A. N. C. S. (2012). Lidar 101 : An Introduction to Lidar Technology , Data , and Applications. *NOAA Coastal Services Center*, pp. 51–76. Obtido de <https://coast.noaa.gov/data/digitalcoast/pdf/lidar-101.pdf>, acedido em dezembro de 2018
- Collin, R. (1985). Dipoles, Arrays and Long-Wire Antennas. Em *Antennas and Radiowave Propagation* (pp. 87–160). McGraw-Hill Book Company.
- Corfield, S. J., & Young, J. M. (2006). Unmanned Surface Vehicles - game changing technology for naval operations. *Advances in Unmanned Marine Vehicles*, pp. 311–328. Obtido de https://digital-library.theiet.org/content/books/10.1049/pbce069e_ch15, acedido em agosto de 2018
- Damas, B. (2019). Protocolos de Comunicações. Em *Arquitetura de Computadores*. [PowerPoint Slides].
- Dubayah, R. O., & Drake, J. B. (2000). Lidar Remote Sensing for Forestry Applications. *Journal of Forestry*, pp. 44–46. Obtido de <http://www.geog.umd.edu/vcl/pubs/jof.pdf>, acedido em novembro de 2018
- El-Rabbany, A. (2002). Introduction to GPS. Em *Introduction to GPS: The Global Positioning System* (pp. 1–11). Artech House. Obtido de <https://books.google.pt/books?id=U2JmghrrB8cC>
- Espressif. (2019). ESP32 Series [Fact sheet]. Obtido de https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf, acedido em janeiro de 2019
- Estado Maior da Armada. (2016). IOA (a definir) – CONCEITO DE EMPREGO OPERACIONAL DE VEÍCULOS NÃO TRIPULADOS SUPERFÍCIE. Lisboa: MARINHA.
- Flir. (2017a). Duo ProR High-Resolution Thermal and Visible-Light Imager for sUAS [Fact sheet]. Obtido de <https://www.flir.com/globalassets/imported-assets/document/17-2249-oem-uas-duo-pro-r-datasheet-final.pdf>, acedido em setembro de 2018
- Flir. (2017b). M100 / M200 SERIES INSTALLATION & OPERATION INSTRUCTIONS [Fact sheet]. Obtido de <https://www.flir.com/globalassets/imported-assets/document/flir-m100-m200-series-user-manual.pdf>, acedido em setembro de 2018
- Flir. (2018). M400 MULTI-SENSOR MARINE THERMAL NIGHT VISION [Fact sheet]. Obtido de <https://www.flir.com/globalassets/imported-assets/document/flir-m400-brochure.pdf>, acedido em setembro de 2018
- Furuno. (2011). GP-320B GPS receiver Operator’s manual [Fact sheet]. Obtido de http://www.furuno.se/fileadmin/files/Manuals/8_GPS_Plotter_ECDIS/GP-320B/GP-320B_OM_ENG_44220C.pdf, acedido em outubro de 2018
- Furuno. (2016). Solid State Doppler Radar DRS4D-NXT [Fact sheet]. Obtido de https://www.furuno.com/files/Brochure/334/upload/DRS4D-NXT_E.pdf, acedido

em outubro de 2018

- Garg, R., Bhartia, P., Bahl, I., & Ittipiboon, A. (2001). *Microstrip Antenna Design Handbook* (1st ed.). Artech House.
- Garmin. (sem data-a). LIDAR-Lite v3HP Operation Manual and Technical Specifications [Fact sheet]. Obtido de https://www.mouser.com/datasheet/2/813/LIDAR_Lite_v3HP_Operation_Manual_and_Technical_Spe-1368030.pdf, acessado em novembro de 2018
- Garmin. (sem data-b). Panoptix PS22-TR [Fact sheet]. Obtido de http://static.garmin.com/pumac/Panoptix_PS22-TR_Install_EN.pdf, acessado em janeiro de 2019
- Garmin. (2012). GPS 19x HVS Technical Specifications [Fact sheet]. Obtido de https://www.fondriest.com/pdf/garmin_19xhvs_spec.pdf, acessado em novembro de 2018
- Garmin. (2014). GMR 18 HD/24 HD Radome Installation Instructions [Fact sheet]. Obtido de http://static.garmin.com/pumac/GMR18-24_18-24_HD_Inst.pdf, acessado em setembro de 2018
- Graham, M. M. (2008). *Unmanned Surface Vehicles : An Operational Commander ' s Tool for Maritime Security*. (Dissertação, Naval War College). Obtido de <https://apps.dtic.mil/dtic/tr/fulltext/u2/a494165.pdf>, acessado em junho de 2018
- Hikvision. (2017). DS-2AE5 IR PTZ Dome Camera [Fact sheet]. Obtido de https://us.hikvision.com/sites/default/files/ds-2ae5123ti-a_data_sheet.pdf, acessado em setembro de 2018
- Hitchcock, R. T. (2012). Physical Characteristics. Em *Radio-Frequency and Microwave Radiation* (pp. 1–9). American Industrial Hygiene Association.
- Hokuyo. (2018). UST-10LX [Fact sheet]. Obtido de <https://www.autonomoustuff.com/wp-content/uploads/2016/07/UST-10LX.pdf>, acessado em setembro de 2018
- Hondex. (2012). HE-7300-Di [Fact sheet]. Obtido de <http://navroc.com/wp-content/uploads/2015/05/Hondex-HE-7300-Di.pdf>, acessado em novembro de 2018
- Huang, Y., & Boyle, K. (2008). Popular Antennas. Em *Antennas - From Theory to Practice* (pp. 129–212). A John Wiley and Sons, Ltd.
- Imagenex. (2004). IMAGENEX MODEL 872 “YELLOWFIN” SIDESCAN SONAR [Fact sheet]. Obtido de https://imagenex.com/assets/images/downloads/YellowFin_v3_Users_Guide.pdf, acessado em novembro de 2018
- International Telecommunication Union – Radiocommunication Sector. (1998). Terms and Definition. *Recommendation ITU-R V.662-2*: Obtido de https://www.itu.int/dms_pubrec/itu-r/rec/v/R-REC-V.662-3-200005-W!!PDF-E.pdf, acessado em dezembro de 2018
- InvenSense. (2016). MPU-9250 Product Specification Revision 1.1 [Fact sheet]. Obtido de <https://www.invensense.com/wp-content/uploads/2015/02/PS-MPU-9250A-01-v1.1.pdf>, acessado em outubro de 2018

- JW Fishers. (2010). SBP-1 Technical Data Sheet [Fact sheet]. Obtido de https://smhttp-ssl-37061.nexcesscdn.net/media/_docs/Datasheets/JWFishers_SBP-1_16_EN.pdf, acedido em novembro de 2018
- Knott, P. (2009). Wire Antenna Modelling with NEC-2. Em *Antenna Engineering* (pp. 1–10). Lawrence Livermore Labs.
- L. Liu, S.W. Cheung, Y. F. W. and T. I. Y. (2012). Cable Effects on Measuring Small Planar UWB Monopole Antennas. *Intech*, pp. 274–294. Obtido de <https://www.intechopen.com/books/ultra-wideband-current-status-and-future-trends/cable-effects-on-measuring-small-planar-uw-b-monopole-antennas>, acedido em fevereiro de 2019
- Lérias, R. (2005). Crónica de Arquitetura Naval. *Anais do Clube Militar Naval*, 735–752. Obtido de http://www.aofa.pt/rimp/ANAIS_VOL_II_2016.pdf, acedido em junho de 2018
- Li, L., Ren, J., & Zhu, Q. (2017). On the application of LoRa LPWAN technology in Sailing Monitoring System. *2017 13th Annual Conference on Wireless On-Demand Network Systems and Services, WONS 2017 - Proceedings*, pp. 77–80. Acedido em Setembro de 2019.
- Liu, Z., Zhang, Y., Yu, X., & Yuan, C. (2016). Unmanned surface vehicles : An overview of developments and challenges. *Annual Reviews in Control*, pp. 1–25. Obtido de https://www.researchgate.net/publication/301831885_Unmanned_surface_vehicles_An_overview_of_developments_and_challenges, acedido em setembro de 2018
- Livox. (2019). Mid Series User Manual [Fact sheet]. Obtido de https://www.livoxtech.com/3296f540ecf5458a8829e01cf429798e/downloads/Livox_Mid_Series_User_Manual_EN_20190118.pdf acedido em novembro de 2018
- Lowrance. (2011). Broadband 3G Radar [Fact sheet]. Obtido de https://ww2.lowrance.com/Root/Lowrance-Documents/Lowrance_Broadband_3G_Radar_en-gb.pdf, acedido em outubro de 2018
- Lowrance. (2018). Point-1 Antenna PRECISION GPS/GLONASS RECEIVER WITH ELECTRONIC COMPASS [Fact sheet]. Obtido de <http://ww2.lowrance.com/en-GB/m/Point-1-en-gb.aspx>, acedido em setembro de 2018
- Lozano-Nieto, A. (1999). Telemetry. Em *The Measurement Instrumentation and Sensors* (pp. 1–5). CRC Press.
- Manley, J. E. (2008). Unmanned surface vehicles, 15 years of development. *OCEANS 2008*, pp. 1–4. Obtido de https://www.researchgate.net/publication/224556921_Unmanned_surface_vehicles_15_years_of_development/link/54e92f620cf25ba91c7eeaab/download, acedido em maio de 2018
- Melvin, W. L., & Scheer, J. A. (2014). Radar Applications. Em *Principles of Modern Radar* (pp. 1–15). Scitech Publishing.
- Microchip. (sem data). LAN8720A Small Footprint RMII 10/100 Ethernet Transceiver [Fact sheet]. Obtido de

- <http://ww1.microchip.com/downloads/en/devicedoc/00002165b.pdf>, acessado em março de 2019
- Moreira, A. (2018). Introdução ao Radar. Em *Sistemas Radar*. [PowerPoint Slides].
- Olex. (2018). Atec 400 - Multibeam Sonar [Fact sheet]. Obtido de https://www.mackaycomm.com/wp-content/uploads/2018/05/Atec-400-Multibeam-sonar-system_web_.pdf, acessado em novembro de 2018
- Parthasarathy, J. (2006). Positioning and Navigation System Using GPS. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science*, pp. 208–212. Obtido de http://www.isprs.org/proceedings/XXXVI/part6/208_XXXVI-part6.pdf, acessado em dezembro de 2018
- Peixeiro, C. (2017). Alguns elementos radiadores. Em *Antenas e Radiopropagação* (pp. 126–145). Instituto Superior Técnico.
- R.Sathishkumar, T.V.S Prasad Gupta, & M.Ajay Babu. (2013). Echo Sounder for Seafloor Object Detection and Classification. *Journal of Engineering, Computers & Applied Sciences*, pp. 32–37. Obtido de <https://pdfs.semanticscholar.org/d850/de02961034553815ec28adc937024c2ede43.pdf>, acessado em fevereiro de 2019
- Raspberry Pi. (2018). Model 3B+ [Fact sheet]. Obtido de <https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-Model-Bplus-Product-Brief.pdf>, acessado em abril de 2019
- Raymarine. (2016). Quantum Radar Installation Instructions [Fact sheet]. Obtido de [https://hudsonmarine.co.uk/uploads/Raymarine/Radar/Quantum Installation instructions.pdf](https://hudsonmarine.co.uk/uploads/Raymarine/Radar/Quantum%20Installation%20instructions.pdf), acessado em outubro de 2018
- RoboSense. (2019). Multi-Beam Real Time LIDAR [Fact sheet]. Obtido de <http://www.high-tech.co.jp/list/wp-content/uploads/RS-LiDAR-16.pdf>, acessado em novembro de 2018
- RPLIDAR. (2016). A1 Low Cost 360 Degree Laser Range Scanner [Fact sheet]. Obtido de <https://www.robotshop.com/media/files/pdf/rplidar-a1m8-360-degree-laser-scanner-development-kit-datasheet-1.pdf>, acessado em novembro de 2018
- Semtech. (2015). SX1276/77/78/79 - Low Power Long Range Transceiver [Fact sheet]. Obtido de https://cdn-shop.adafruit.com/product-files/3179/sx1276_77_78_79.pdf, acessado em fevereiro de 2019
- Seybold, J. S. (2005). Antenna Fundamentals. Em *Introduction to RF Propagation* (pp. 38–65). A John Wiley & Sons, INC.
- Sharma, N. (2017). A Study Of Different Feeding Mechanisms In Microstrip Patch Antenna. *International Journal of Microwaves Applications*, pp. 1–9. Obtido de https://www.researchgate.net/publication/315113944_A_Study_Of_Different_Feeding_Mechanisms_In_Microstrip_Patch_Antenna, acessado em janeiro de 2019
- Skolnik, M. I. (1990). An Introduction to Radar. Em *Radar Handbook* (pp. 1–18). McGraw Hill.
- STMicroelectronics. (2015). LSM9DS1 iNEMO inertial module [Fact sheet]. Obtido de

- <https://www.st.com/resource/en/datasheet/DM00103319.pdf>, acessado em outubro de 2018
- Subsea Tech. (2015). CAT-Surveyor [Brochure]. Obtido de <https://www.subsea-tech.com/cat-surveyor/>, acessado em junho de 2018
- u-blox. (2016). NEO-M8 concurrent GNSS modules [Fact sheet]. Obtido de https://www.u-blox.com/sites/default/files/NEO-M8_DataSheet_%28UBX-13003366%29.pdf, acessado em abril de 2019
- Velleman. (2017). SD CARD LOGGING SHIELD VMA304 [Fact sheet]. Obtido de https://www.velleman.eu/downloads/29/vma304_a4v01.pdf, acessado em maio de 2019
- Waveshare. (2013). RS232 board [Fact sheet]. Obtido de https://www.waveshare.com/wiki/RS232_Board, acessado em abril de 2019
- Woodard, C. (2008). Caribbean Pirates and the Man Who Brought Them Down. Em *The Republic of Pirates* (pp. 139–207). HMH Books.
- Younes, B. (2015). *Robotização de uma Cápsula para salvamento marítimo Abordagem numa Perspetiva de Arquitetura Naval*. Repositório Comum - Escola Naval. Obtido de <http://hdl.handle.net/10400.26/10856>, acessado em junho de 2018
- YSI. (2019a). EXO User Manual ADVANCED WATER QUALITY MONITORING PLATFORM [Fact sheet]. Obtido de <https://www.ysi.com/FileLibrary/Documents/Manuals/EXO-User-Manual-Web.pdf>, acessado em junho de 2018
- YSI. (2019b). HYCAT - Autonomous Surface Vehicle [Fact sheet]. Obtido de <https://www.ysi.com/FileLibrary/Documents/SpecificationSheets/HYCAT-Autonomous-Surface-Vehicle-Specification-Sheet.pdf>, acessado em junho de 2018
- Yuh, J., Marani, G., & Richard Blidberg, D. (2011). Applications of marine robotic vehicles. *Intelligent Service Robotics*, pp. 221–231. Obtido de https://www.researchgate.net/publication/226028601_Applications_of_marine_robotic_vehicles, acessado em setembro de 2018

Apêndice A – Código Protocolo RS232

```
// Protocolo RS232 -> ESP32 //
```

```
/*-----*/  
//BIBLIOTECAS//  
#include <SPI.h> //protocolo SPI  
#include <LoRa.h> //comms chip SX1276 - LoRa (433MHz)  
#include <TinyGPS++.h> //modulo GPS - ublox M8N  
#include <Wire.h> //protocolo I2C  
#include <Adafruit_Sensor.h> //camada data-logging para sensores  
#include <Adafruit_BNO055.h> //modulo IMU - BOSCH BNO055  
#include <utility/imumaths.h> //cálculos da fusao sensorial  
#include <WiFi.h> //protocolo comms WiFi 802.11 (2.4GHz)  
/*-----*/  
#define INIT_C 0x2B //código de inicio de mensagem  
#define TIMEOUT 2000 //prevenção de bloqueio da rotina  
#define MAX_MSG_LENGTH 64  
/*-----*/  
#define CMD_GPS_LATITUDE 0  
#define CMD_GPS_LONGITUDE 1  
#define CMD_GPS_DATA 2  
#define CMD_GPS_TIME_UTC 3 //atribuição de valores a variáveis  
#define CMD_GPS_TIME_HORA 4 //variáveis -> código de mensagem  
#define CMD_GPS_TIME_MINUTOS 5 //protocolo RS232  
#define CMD_GPS_TIME_SEGUNDOS 6 //origem do comando -> raspberry  
#define CMD_GPS_SATELITES 7 //pedido de dados  
#define CMD_GPS_SOG 8  
#define CMD_GPS_COG 9  
#define CMD_IMU_HEADING 10  
#define CMD_NMEA_MESSAGE 11  
#define CMD_SENSOR_INFO 12  
/*-----*/  
#define REPLY_GPS_LATITUDE 0  
#define REPLY_GPS_LONGITUDE 1  
#define REPLY_GPS_DATA 2  
#define REPLY_GPS_TIME_UTC 3 //atribuição de valores a variáveis  
#define REPLY_GPS_TIME_HORA 4 //variáveis -> código de mensagem  
#define REPLY_GPS_TIME_MINUTOS 5 //protocolo RS232  
#define REPLY_GPS_TIME_SEGUNDOS 6 //resposta ao pedido de dados  
#define REPLY_GPS_SATELITES 7  
#define REPLY_GPS_SOG 8  
#define REPLY_GPS_COG 9  
#define REPLY_IMU_HEADING 10  
#define REPLY_NMEA_MESSAGE 11  
#define REPLY_SENSOR_INFO 12  
#define REPLY_TRANSMISSION_ERROR 13  
#define REPLY_TIMEOUT 14  
/*-----*/  
#define SCK 5  
#define MISO 19
```

```

#define MOSI      27                //pins de ligação SPI SX1276
#define SS        18
#define RST       14
#define DIO       26
#define BAND      433E6             //banda de frequência -> 433 MHz
#define PABOOST true                //escolha do amplificador de potência
                                   //(chip sx1276 possui dois amplificadores -> 20 ou 14 dBm)
/*-----*/
typedef enum {WAITING, READ_DATASIZE, READ_COMMAND, READ_DATA,
READ_CHKSUM} CommState;
CommState state;
               //definição do processo sequencial do estado de
               //receção de msg encapsulada
/*-----*/
uint8_t byteReceived;
uint8_t message[MAX_MSG_LENGTH];
uint8_t dataSize, dataCount;
uint8_t replyMessage[MAX_MSG_LENGTH];
float r = M_PI / 180.0f;
float d = 180.0f / M_PI;
float magd = -72.4;
const char* ssid      = "VENTSUPEN";
const char* password = "vieirarodrigues";
const char* host = "192.168.4.1";
float rawHeading;
float heading;
float hdt;
float latitude;
float longitude;
int latitudeD;
float latitudeDM;
int longitudeD;
float longitudeDM;
int longitudeD0;
String utc;
String equador;
String meridiano;
String satelites;
String sog;
String cog;
String ddate;
String hora;
String minutos;
String segundos;
int httpPort;
int packetSize;
String outgoing;
long lastSendTime = 0;
int interval = 500;
char a;
unsigned long t0;
unsigned int counter = 0;
String nmeaStr;
String sensorStr;
bool checksumXOR();
uint8_t calculateChecksumXOR(const uint8_t* data, uint16_t n, uint8_t
chkSum = 0);
/*-----*/

```

```

TinyGPSPlus gps;
Adafruit_BNO055 bno = Adafruit_BNO055(55);
/*-----*/
void setup() {
  message[0] = INIT_C;          //definição do simbolo de inicio de msg
  replyMessage[0] = INIT_C;
  state = WAITING;              //Estado inicial da sequência
  t0 = millis();               //de recepção de msg
  Serial.begin(115200, SERIAL_8N1, 3, 1); //iniciar porta série
  Serial1.begin(9600, SERIAL_8N1, 12, 15); //iniciar GPS
  Serial2.begin(115200);        //iniciar monitor série
  SPI.begin(SCK, MISO, MOSI, SS); //iniciar LoRa
  WiFi.begin(ssid, password);   //iniciar WiFi
  LoRa.setPins(SS, RST, DI0);
  if (!LoRa.begin(BAND, PABOOST)) {
    Serial2.print("LoRa fail!");
  }
  while (!Serial);
    Serial2.print("IMU fail!");
  {
    while (1);
  }
  bno.setExtCrystalUse(true);
  while (WiFi.status() != WL_CONNECTED) { //tempo de espera para
    delay(500);                          //se conectar a um cliente
  }
}
/*-----*/
void send2G4() { //Transmissão de dados por WiFi
  WiFiClient client; //Procura e liga-se ao cliente
  httpPort = 80;
  if (!client.connect(host, httpPort)) {
    Serial.println("connection failed");
    return; //Se o cliente estiver disponível envia-se
  } //todas as informações dos sensores
  client.print(sensorStr); //Enviar msg recebida do raspberry
  while (client.available()) {
    String line = client.readStringUntil('\r');
    Serial.print(line); //Estando ligado ao cliente
  } // "Ouve" qualquer mensagem recebida
}
/*-----*/
void send433() { //Transmissão de pacotes com dados
  LoRa.beginPacket(); //do GPS por LoRa em formato NMEA
  LoRa.print(nmeaStr); //Enviar msg recebida do raspberry
  LoRa.endPacket();
}
/*-----*/
void loop() {
  if (Serial.available() <= 0) {
    if ( state != WAITING && (millis() - t0) > TIMEOUT ) {
      replyMessage[1] = 1; // Ddatasize = 0
      replyMessage[2] = REPLY_TIMEOUT; //se não receber nenhum
      //comando dentro do timeout,
      sendReplyMessage(); //envia msg de erro ao rpi
      state = WAITING; //e volta para o estado de espera
    }
    return;
  }
}

```

```

}
int val = Serial.read();
if ( val == -1)                //guardar bytes recebidos numa variável
    return;                    //quando recebe msg
else
    byteReceived = byte(val);

switch (state) {
case WAITING:
    if ( byteReceived == INIT_C ) {
        state = READ_DATASIZE;
        t0 = millis(); // Start receiving message
    }
    break;
case READ_DATASIZE:                //caso de escolha múltipla
    message[1] = byteReceived;      //para descodificação da
    dataSize = byteReceived - 1;    //mensagem
    state = READ_COMMAND;
    break;
case READ_COMMAND:
    message[2] = byteReceived;
    if ( dataSize == 0)
        state = READ_CHKSUM;
    else
    {
        state = READ_DATA;
        dataCount = 0;
    }
    break;
case READ_DATA:
    message[3 + dataCount] = byteReceived;
    if ( ++dataCount == dataSize )
        state = READ_CHKSUM;
    break;
case READ_CHKSUM:
    message[3 + dataSize] = byteReceived;
    processMessage();
    sendReplyMessage();
    state = WAITING;
    break;
}
}
/*-----*/
bool checkSumXOR() {
    return (calculateCheckSumXOR(message, 3 + dataSize) == message[3 +
dataSize]);
}
uint8_t calculateCheckSumXOR(const uint8_t* data, uint16_t n, uint8_t
chkSum) {
    while (n--)
        chkSum ^= *data++;          //cálculo do checksum
    return chkSum;
}
/*-----*/
void processMessage() {
    if ( !checkSumXOR() ) { // Transmission Error!
        replyMessage[2] = REPLY_TRANSMISSION_ERROR;
        return;              //Quando o algoritmo de verificação

```



```

    } //deteta mensagens corrompidas, envia msg de erro
/*-----*/
switch (message[2]) { //Command
    case CMD_GPS_LATITUDE:
        readLatitude();
        break;
    case CMD_GPS_LONGITUDE:
        readLongitude();
        break;
    case CMD_GPS_DATA: //depois de decodificada a msg
                        //a partir do código da msg
        readData(); //determina qual a função que deve correr
        break;
    case CMD_GPS_TIME_UTC:
        readUTC();
        break;
    case CMD_GPS_TIME_HORA:
        readHora();
        break;
    case CMD_GPS_TIME_MINUTOS:
        readMinutos();
        break;
    case CMD_GPS_TIME_SEGUNDOS:
        readSegundos();
        break;
    case CMD_GPS_SATELITES:
        readSatelites();
        break;
    case CMD_GPS_SOG:
        readSOG();
        break;
    case CMD_GPS_COG:
        readCOG();
        break;
    case CMD_IMU_HEADING:
        readHeading();
        break;
    case CMD_NMEA_MESSAGE:
        readNMEA();
        send433();
        break;
    case CMD_SENSOR_INFO:
        readINFO();
        send2G4();
        break;
}
}
/*-----*/
void sendReplyMessage() {
    replyMessage[2 + replyMessage[1]] = '\n';
    Serial.write(replyMessage, 3 + replyMessage[1]);
    Serial.flush();
} //contrução e transmissão da
mensagem de resposta
/*-----*/
static void smartDelay(unsigned long ms)
{
    unsigned long start = millis(); //código de verificação
    do //posição GPS arualizada

```

```

    {
        while (Serial1.available())
            gps.encode(Serial1.read());
    } while (millis() - start < ms);
}
/*-----*/
void readINFO() {
    for (int i = 0; i < dataSize; i++) {
        sensorStr[i] = message[i + 3];
    }
    //construção de msg
    //a enviar por WiFi, vinda do
    //raspberry
}
/*-----*/
void readNMEA() {
    for (int i = 0; i < dataSize; i++) {
        nmeaStr[i] = message[i + 3];
    }
    //construção de msg
    //a enviar por LoRa, vinda do
    //raspberry
}
}
}
/*-----*/
void readLatitude() {
    latitude = gps.location.lat(); //leitura da latitude e
    if (latitude < 0) { //transformação em formato NMEA
        latitude = latitude * -1;
    }
    latitudeD = floor(latitude);
    latitudeDM = ((latitude - latitudeD) * 60);
    equador = (gps.location.rawLng().negative ? "N" : "S");
    String finalLatitude = String(latitudeD) + String(latitudeDM, 5) +
    "," + (equador);

    int replySize1 = finalLatitude.length();
    replyMessage[2] = REPLY_GPS_LATITUDE; //cálculo do tamanho
    replyMessage[1] += replySize1; //da mensagem e construção
    for (int i = 0; i < replySize1; i++) { //de array dos dados a
        replyMessage[i + 3] = finalLatitude.charAt(i); //serem enviados
    }
}
/*-----*/
void readLongitude() {
    longitude = gps.location.lng(); //leitura da longitude e
    if (longitude < 0) { //transformação em formato NMEA
        longitude = longitude * -1;
    }
    longitudeD = floor(longitude);
    longitudeD0 = longitudeD * 10;
    longitudeDM = ((longitude - longitudeD) * 60);
    meridiano = (gps.location.rawLng().negative ? "W" : "E");
    String finalLongitude = String(longitudeD0) + String(longitudeDM, 5)
+ "," + (meridiano);

    int replySize2 = finalLongitude.length();
    replyMessage[2] = REPLY_GPS_LONGITUDE; //cálculo do tamanho
    replyMessage[1] += replySize2; //da mensagem e construção
    for (int i = 0; i < replySize2; i++) { //de array dos dados a
        replyMessage[i + 3] = finalLongitude.charAt(i); //serem enviados
    }
}
}

```

```

/*-----*/
oid readData() {
    ddate = (gps.date.value());          //leitura da data
    String finalData = (ddate);

    int replySize3 = finalData.length();
    replyMessage[2] = REPLY_GPS_DATA;      //cálculo do tamanho
    replyMessage[1] += replySize3;        //da mensagem e construção
    for (int i = 0; i < replySize3; i++) { //de array dos dados a
        replyMessage[i + 3] = finalData.charAt(i); //serem enviados
    }
}
/*-----*/
void readUTC() {
    utc = (gps.time.value()) / 100;
    String finalUTC = (utc);              //leitura de registo horário
                                          //(formato: hhmmss)

    int replySize4 = finalUTC.length();
    replyMessage[2] = REPLY_GPS_TIME_UTC; //cálculo do tamanho
    replyMessage[1] += replySize4;        //da mensagem e construção
    for (int i = 0; i < replySize4; i++) { //de array dos dados a
        replyMessage[i + 3] = finalUTC.charAt(i); //serem enviados
    }
}
/*-----*/
void readHora() {
    hora = (gps.time.hour());
    String finalHora = (hora);           //leitura da hora

    int replySize5 = finalHora.length();
    replyMessage[2] = REPLY_GPS_TIME_HORA; //cálculo do tamanho
    replyMessage[1] += replySize5;        //da mensagem e construção
    for (int i = 0; i < replySize5; i++) { //de array dos dados a
        replyMessage[i + 3] = finalHora.charAt(i); //serem enviados
    }
}
/*-----*/
void readMinutos() {
    minutos = (gps.time.minute());
    String finalMinutos = (minutos);     //leitura dos minutos

    int replySize6 = finalMinutos.length();
    replyMessage[2] = REPLY_GPS_TIME_MINUTOS; //cálculo do tamanho
    replyMessage[1] += replySize6;        //da mensagem e construção
    for (int i = 0; i < replySize6; i++) { //de array dos dados a
        replyMessage[i + 3] = finalMinutos.charAt(i); //serem enviados
    }
}
/*-----*/
void readSegundos() {
    segundos = (gps.time.second());
    String finalSegundos = (segundos);    //leitura dos segundos

    int replySize7 = finalSegundos.length();
    replyMessage[2] = REPLY_GPS_TIME_SEGUNDOS; //cálculo do tamanho
    replyMessage[1] += replySize7;        //da mensagem e construção
    for (int i = 0; i < replySize7; i++) { //de array dos dados a
        replyMessage[i + 3] = finalSegundos.charAt(i); //serem enviados
    }
}

```

```

    }
}
/*-----*/
void readSatellites() {
    satellites = (gps.satellites.value()); //leitura do numero de
    String finalSatellites = (satellites); //satellites disponiveis

    int replySize8 = finalSatellites.length();
    replyMessage[2] = REPLY_GPS_SATELITES; //cálculo do tamanho
    replyMessage[1] += replySize8; //da mensagem e construção
    for (int i = 0; i < replySize8; i++) { //de array dos dados a
        replyMessage[i + 3] = finalSatellites.charAt(i); //serem enviados
    }
}
/*-----*/
void readSOG() {
    sog = (gps.speed.knots()); //leitura da velocidade em nós
    String finalSOG = (sog);

    int replySize9 = finalSOG.length();
    replyMessage[2] = REPLY_GPS_SOG; //cálculo do tamanho
    replyMessage[1] += replySize9; //da mensagem e construção
    for (int i = 0; i < replySize9; i++) { //de array dos dados a
        replyMessage[i + 3] = finalSOG.charAt(i); //serem enviados
    }
}
/*-----*/
void readCOG() {
    cog = (gps.course.deg());
    String finalCOG = (cog); //leitura do rumo

    int replySize10 = finalCOG.length();
    replyMessage[2] = REPLY_GPS_COG; //cálculo do tamanho
    replyMessage[1] += replySize10; //da mensagem e construção
    for (int i = 0; i < replySize10; i++) { //de array dos dados a
        replyMessage[i + 3] = finalCOG.charAt(i); //serem enviados
    }
}
/*-----*/
void readHeading() {
    sensors_event_t event;
    bno.getEvent(&event);
    rawHeading = (event.orientation.x); //leitura da orientação
    heading = rawHeading - 90; //no eixo xx (horizontal)
    if (rawHeading < 90 && rawHeading >= -179.99) {
        heading = rawHeading + 270;
    }
    hdt = heading - magd; //conversão da orientação obtida
    if (hdt >= 360) { //pelo IMU em leituras de proa
        hdt = hdt - 360; //simulando uma bussola compensada
    } //por valores do giroscopio
    if (hdt < 0) {
        hdt = hdt + 360;
    }
    int compass = floor(hdt);
    int decimal = (hdt - compass) * 100;
    String finalHeading = String(compass) + "." + String(decimal);
}

```

```

int replySize11 = finalHeading.length();
replyMessage[2] = REPLY_IMU_HEADING;          //cálculo do tamanho
replyMessage[1] += replySize11;                //da mensagem e construção
for (int i = 0; i < replySize11; i++) {        //de array dos dados a
    replyMessage[i + 3] = finalHeading.charAt(i); //serem enviados
}
}
/*-----*/

```


Apêndice B – Código *ESP32*

```
// VENT-SUP-EN -> ESP32 //
```

```
/*-----*/  
//BIBLIOTECAS//  
#include <SPI.h> //protocolo SPI  
#include <LoRa.h> //comms chip SX1276 - LoRa (433MHz)  
#include <TinyGPS++.h> //modulo GPS - ublox M8N  
#include <Wire.h> //protocolo I2C  
#include <Adafruit_Sensor.h> //camada data-logging para sensores  
#include <Adafruit_BNO055.h> //modulo IMU - BOSCH BNO055  
#include <utility/imumaths.h> //cálculos da fusao sensorial  
#include <WiFi.h> //protocolo comms WiFi 802.11 (2.4GHz)  
/*-----*/  
#define INIT_C 0x2B //código de início de mensagem  
#define TIMEOUT 2000 //prevenção de bloqueio da rotina  
#define MAX_MSG_LENGTH 64  
/*-----*/  
#define REPLY_GPS_LATITUDE 0  
#define REPLY_GPS_LONGITUDE 1  
#define REPLY_GPS_TIME_UTC 2  
#define REPLY_GPS_SOG 3 //atribuição de valores a variáveis  
#define REPLY_GPS_COG 4 //variáveis -> código de mensagem  
#define REPLY_IMU_HEADING 5 //protocolo RS232  
#define REPLY_NMEA_MESSAGE 6  
#define REPLY_SENSOR_INFO 7  
#define REPLY_GPS_SATELITES 8  
#define REPLY_GPS_DATA 9  
#define REPLY_GPS_TIME_HORA 10  
#define REPLY_GPS_TIME_MINUTOS 11  
#define REPLY_GPS_TIME_SEGUNDOS 12  
#define REPLY_TRANSMISSION_ERROR 13  
#define REPLY_TIMEOUT 14  
/*-----*/  
#define SCK 5  
#define MISO 19  
#define MOSI 27 //pins de ligação SPI SX1276  
#define SS 18  
#define RST 14  
#define DI0 26  
#define BAND 433E6 //banda de frequência -> 433 MHz  
#define PABOOST true //escolha do amplificador de potência  
// (chip sx1276 possui dois amplificadores -> 20 ou 14 dBm)  
/*-----*/  
uint8_t byteReceived;  
uint8_t message[MAX_MSG_LENGTH];  
uint8_t dataSize, dataCount;  
uint8_t replyMessage[MAX_MSG_LENGTH];  
float r = M_PI / 180.0f;  
float d = 180.0f / M_PI;  
float magd = -72.4;  
const char* ssid = "VENTSUPEN";  
const char* password = "vieirarodrigues";  
const char* host = "192.168.4.1";
```

```

float rawHeading;
float heading;
float hdt;
float latitude;
float longitude;
int latitudeD;
float latitudeDM; //declaração de variáveis
int longitudeD;
int longitudeD0;
float longitudeDM;
String utc;
String equador;
String meridiano;
String satelites;
String sog;
String cog;
String ddate;
String hora;
String minutos;
String segundos;
int httpPort;
int packetSize;
String outgoing;
long lastSendTime = 0;
int interval = 500;
char a;
unsigned long t0;
unsigned int counter = 0;
String nmeaStr;
String sensorStr;
bool checksumXOR();
uint8_t calculateChecksumXOR(const uint8_t* data, uint16_t n, uint8_t
chkSum = 0);
/*-----*/
TinyGPSPlus gps;
Adafruit_BNO055 bno = Adafruit_BNO055(55);
/*-----*/
void setup() {
  message[0] = INIT_C;
  replyMessage[0] = INIT_C;
  t0 = millis(); //inicialização
  Serial.begin(115200, SERIAL_8N1, 3, 1); //de todos os
  Serial1.begin(9600, SERIAL_8N1, 12, 15); //componentes
  Serial2.begin(115200);
  SPI.begin(SCK, MISO, MOSI, SS);
  WiFi.begin(ssid, password);
  LoRa.setPins(SS, RST, DI0);
  if (!LoRa.begin(BAND, PABOOST)) {
    Serial2.print("LoRa fail!");
  }
  if (!bno.begin()){
    Serial2.print("IMU fail!");
  }
  bno.setExtCrystalUse(true);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
  }
}

```



```

/*-----*/
void send2G4() { //Transmissão de dados por WiFi
    WiFiClient client; //Procura, escolhe e liga-se ao cliente
    httpPort = 80;
    if (!client.connect(host, httpPort)) {
        Serial.println("connection failed");
        return; //Se o cliente estiver disponível envia-se
    } //todas as informações dos sensores
    client.print(String("Hora -> ") + hora + String(":") + minutos +
String(":") + segundos + "\n" + String("Latitude -> ") + latitudeD +
String("°") + latitudeDM + equador + "\n" + String("Longitude -> ") +
longitudeD + String("°") + longitudeDM + meridiano + "\n" +
String("Proa -> ") + hdt + String("°") + "\n" + String("COG -> ") +
cog + String("°") + "\n" + String("SOG -> ") + sog + String(" nós") +
"\n" + String("-----") + "\n" + "\r");
    while (client.available()) {
        String line = client.readStringUntil('\r');
        Serial2.print(line); //Estando ligado ao cliente
    } //Ouve qualquer mensagem recebida
}
/*-----*/
void send433_1() { //Transmissão de pacotes com dados
    LoRa.beginPacket(); //do GPS por LoRa em formato NMEA
    LoRa.print("$GPGGA," + String(utc) + "," + String(latitudeD) +
String(latitudeDM, 5) + "," + String(equador) + "," +
String(longitudeD0) + String(longitudeDM, 5) + "," + String(meridiano)
+ ",1," + String(satelites) + "\n");
    LoRa.endPacket();
}
/*-----*/
void send433_2() { //Transmissão de pacotes com dados
    LoRa.beginPacket(); //do GPS por LoRa em formato NMEA
    LoRa.print("GPRMC," + String(utc) + ",A," + String(latitudeD) +
String(latitudeDM, 5) + "," + String(equador) + "," +
String(longitudeD0) + String(longitudeDM, 5) + "," + String(meridiano)
+ "," + String(sog) + "," + String(cog) + "," + String(ddate) + "\n");
    LoRa.endPacket();
}
/*-----*/
void send433_3() { //Transmissão de pacotes com dados
    LoRa.beginPacket(); //do IMU por LoRa em formato NMEA
    LoRa.print("$HCHDT," + String(hdt) + "\n");
    LoRa.endPacket();
}
/*-----*/
void loop() { //Rotina principal do microprocessador
    if (Serial.available() <= 0) { //Todas as funções são chamadas
        if ((millis() - t0) > TIMEOUT) { //a partir desta rotina
            replyMessage[1] = 1; // Ddatasize = 0
            replyMessage[2] = REPLY_TIMEOUT; //Se ocorrer algum
            //problema, o ESP32 envia
            sendReplyMessage(); //mensagem de erro ao RPI
        }
        return; //Enquanto a porta serie estiver
    } //disponível o ESP32 irá responder
    if (Serial.available()) { //aos comandos/pedidos enviados
        cmd(Serial.read() - '0'); //pelo Raspberry
    }
}

```

```

}
/*-----*/
void cmd(int n) {
    switch (n) {          //consoante o código da mensagem recebido
        case 1:           //irá ser chamada uma função correspondente
            readLatitude(); //ao pedido fazendo uma leitura e
            sendReplyMessage(); //posterior dos transmissão dos dados
            break;         //ou uma transmissão destes para a
        case 2:           //estação em terra na banda desejada
            readLongitude();
            sendReplyMessage();
            break;
        case 3:
            readUTC();
            sendReplyMessage();
            break;
        case 4:
            readSOG();
            sendReplyMessage();
            break;
        case 5:
            readCOG();
            sendReplyMessage();
            break;
        case 6:
            readHeading();
            sendReplyMessage();
            break;
        case 7:
            send433_1();
            break;
        case 8:
            send2G4();
            break;
        case 9:
            send433_2();
            break;
        case 10:
            send433_3();
            break;
    }
}
/*-----*/
bool checksumXOR() {
    return (calculateChecksumXOR(message, 3 + dataSize) == message[3 +
dataSize]);
}
uint8_t calculateChecksumXOR(const uint8_t* data, uint16_t n, uint8_t
chkSum) {
    while (n--)
        chkSum ^= *data++;          //cálculo do checksum

    return chkSum;
}
/*-----*/
void sendReplyMessage() {
    replyMessage[2 + replyMessage[1]] =
calculateChecksumXOR(replyMessage, 2 + replyMessage[1]);
}

```

```

        replyMessage[3 + replyMessage[1]] = '\n';
        Serial.write(replyMessage, 4 + replyMessage[1]);
        Serial.flush();
    } //contrução e transmissão da mensagem de resposta
    /*-----*/
    static void smartDelay(unsigned long ms)
    {
        unsigned long start = millis(); //código de verificação
    de
        {
            while (Serial1.available()) //posição de GPS atualizada
            {
                gps.encode(Serial1.read());
            } while (millis() - start < ms);
        }
    }
    /*-----*/
    void readLatitude() {
        latitude = gps.location.lat(); //leitura da latitude e
        if (latitude < 0) { //transformação em formato NMEA
            latitude = latitude * -1;
        }
        latitudeD = floor(latitude);
        latitudeDM = ((latitude - latitudeD) * 60);
        equador = (gps.location.rawLng().negative ? "N" : "S");
        String finalLatitude = String(latitudeD) + String(latitudeDM, 5) +
        ", " + (equador);

        int replySize1 = finalLatitude.length();
        replyMessage[2] = REPLY_GPS_LATITUDE; //cálculo do tamanho
        replyMessage[1] += replySize1; //da mensagem e construção
        for (int i = 0; i < replySize1; i++) { //de array dos dados a
            replyMessage[i + 3] = finalLatitude.charAt(i); //serem enviados
        }
    }
    /*-----*/
    void readLongitude() {
        longitude = gps.location.lng(); //leitura da longitude e
        if (longitude < 0) { //transformação em formato NMEA
            longitude = longitude * -1;
        }
        longitudeD = floor(longitude);
        longitudeD0 = longitudeD * 10;
        longitudeDM = ((longitude - longitudeD) * 60);
        meridiano = (gps.location.rawLng().negative ? "W" : "E");
        String finalLongitude = String(longitudeD0) + String(longitudeDM, 5)
+ ", " + (meridiano);

        int replySize2 = finalLongitude.length();
        replyMessage[2] = REPLY_GPS_LONGITUDE; //cálculo do tamanho
        replyMessage[1] += replySize2; //da mensagem e construção
        for (int i = 0; i < replySize2; i++) { //de array dos dados a
            replyMessage[i + 3] = finalLongitude.charAt(i); //serem enviados
        }
    }
    /*-----*/
    void readData() {
        ddate = (gps.date.value()); //leitura da data
        String finalData = (ddate);
    }

```

```

    int replySize3 = finalData.length();
    replyMessage[2] = REPLY_GPS_DATA;           //cálculo do tamanho
    replyMessage[1] += replySize3;              //da mensagem e construção
    for (int i = 0; i < replySize3; i++) {      //de array dos dados a
        replyMessage[i + 3] = finalData.charAt(i); //serem enviados
    }
}
/*-----*/
void readUTC() {
    utc = (gps.time.value()) / 100;
    String finalUTC = (utc);                   //leitura de registo horário
                                              //(formato: hhmmss)

    int replySize4 = finalUTC.length();
    replyMessage[2] = REPLY_GPS_TIME_UTC;       //cálculo do tamanho
    replyMessage[1] += replySize4;              //da mensagem e construção
    for (int i = 0; i < replySize4; i++) {      //de array dos dados a
        replyMessage[i + 3] = finalUTC.charAt(i); //serem enviados
    }
}
/*-----*/
void readSatellites() {
    satellites = (gps.satellites.value());      //leitura do numero de
    String finalSatellites = (satellites);       //satellites disponiveis

    int replySize8 = finalSatellites.length();
    replyMessage[2] = REPLY_GPS_SATELITES;      //cálculo do tamanho
    replyMessage[1] += replySize8;              //da mensagem e construção
    for (int i = 0; i < replySize8; i++) {      //de array dos dados a
        replyMessage[i + 3] = finalSatellites.charAt(i); //serem enviados
    }
}
/*-----*/
void readSOG() {
    sog = (gps.speed.knots());                  //leitura da velocidade em nós
    String finalSOG = (sog);

    int replySize9 = finalSOG.length();
    replyMessage[2] = REPLY_GPS_SOG;            //cálculo do tamanho
    replyMessage[1] += replySize9;              //da mensagem e construção
    for (int i = 0; i < replySize9; i++) {      //de array dos dados a
        replyMessage[i + 3] = finalSOG.charAt(i); //serem enviados
    }
}
/*-----*/
void readCOG() {
    cog = (gps.course.deg());
    String finalCOG = (cog);                   //leitura do rumo

    int replySize10 = finalCOG.length();
    replyMessage[2] = REPLY_GPS_COG;           //cálculo do tamanho
    replyMessage[1] += replySize10;            //da mensagem e construção
    for (int i = 0; i < replySize10; i++) {    //de array dos dados a
        replyMessage[i + 3] = finalCOG.charAt(i); //serem enviados
    }
}
/*-----*/
void readHeading() {
    sensors_event_t event;

```

```

bno.getEvent(&event);
rawHeading = (event.orientation.x);    //leitura da orientação
heading = rawHeading - 90;             //no eixo xx (horizontal)
if (rawHeading < 90 && rawHeading >= -179.99) {
    heading = rawHeading + 270;
}
hdt = heading - magd;                  //conversão da orientação obtida
if (hdt >= 360) {                      //pelo IMU em leituras de proa
    hdt = hdt - 360;                   //simulando uma bussola compensada
}                                     //por valores do giroscópio
if (hdt < 0) {
    hdt = hdt + 360;
}
int compass = floor(hdt);
int decimal = (hdt - compass) * 100;
String finalHeading = String(compass) + "." + String(decimal);

int replySize11 = finalHeading.length();
replyMessage[2] = REPLY_IMU_HEADING;    //cálculo do tamanho
replyMessage[1] += replySize11;         //da mensagem e construção
for (int i = 0; i < replySize11; i++) { //de array dos dados a
    replyMessage[i + 3] = finalHeading.charAt(i); //serem enviados
}
}
/*-----*/
void readHora() {
    hora = (gps.time.hour());
    String finalHora = (hora);           //leitura da hora

    int replySize5 = finalHora.length();
    replyMessage[2] = REPLY_GPS_TIME_HORA;    //cálculo do tamanho
    replyMessage[1] += replySize5;           //da mensagem e construção
    for (int i = 0; i < replySize5; i++) {    //de array dos dados a
        replyMessage[i + 3] = finalHora.charAt(i); //serem enviados
    }
}
/*-----*/
void readMinutos() {
    minutos = (gps.time.minute());
    String finalMinutos = (minutos);         //leitura dos minutos

    int replySize6 = finalMinutos.length();
    replyMessage[2] = REPLY_GPS_TIME_MINUTOS; //cálculo do tamanho
    replyMessage[1] += replySize6;           //da mensagem e construção
    for (int i = 0; i < replySize6; i++) {    //de array dos dados a
        replyMessage[i + 3] = finalMinutos.charAt(i); //serem enviados
    }
}
/*-----*/
void readSegundos() {
    segundos = (gps.time.second());
    String finalSegundos = (segundos);       //leitura dos segundos

    int replySize7 = finalSegundos.length();
    replyMessage[2] = REPLY_GPS_TIME_SEGUNDOS; //cálculo do tamanho
    replyMessage[1] += replySize7;           //da mensagem e construção
    for (int i = 0; i < replySize7; i++) {    //de array dos dados a
        replyMessage[i + 3] = finalSegundos.charAt(i); //serem enviados
    }
}

```

```
    }  
}  
/*-----*/
```

Apêndice C – Código *Raspberry Pi*

```
# VENT-SUP-EN -> Raspberry Pi #

#-----#
#BIBLIOTECAS#
import time #tempo
import serial #protocolo RS232
from XOR_CheckSum import xor_checksum_string #cálculo checksum
#-----#
ser = serial.Serial("/dev/ttyS0", 115200) #ligação RS232
ser.timeout = 5
#-----#
def latitude(): #função de leitura dos dados de latitude
    ser.write('1'.encode()) #transmissão código de pedido/comando
    data1 = ser.readline() #leitura dos dados
    list1 = list(data1) #conversão dos dados para um array
    global latitude #declaração da variável
    global in_checksum1 #declaração da variável
    global out_checksum1 #declaração da variável
    latitude = bytes(list1[3:15]) #criação de array de bytes
    in_checksum1 = hex(xor_checksum_string(bytes(list1[0:15])))
    out_checksum1 = list1[16] #cálculo de checksum da
    #mensagem de latitude
    #e gravação em variável do checksum vindo do ESP32
#-----#
def longitude(): #função de leitura dos dados de longitude
    ser.write('2'.encode()) #transmissão código de pedido/comando
    data2 = ser.readline() #leitura dos dados
    list2 = list(data2) #conversão dos dados para um array
    global longitude #declaração da variável
    global in_checksum2 #declaração da variável
    global out_checksum2 #declaração da variável
    longitude = bytes(list2[3:14]) #criação de array de bytes
    in_checksum2 = hex(xor_checksum_string(bytes(list2[0:14])))
    out_checksum2 = list2[15] #cálculo de checksum da
    #mensagem de latitude
    #e gravação em variável do checksum vindo do ESP32
#-----#
def UTC(): #função de leitura dos dados de registo horário
    ser.write('3'.encode()) #transmissão código de pedido/comando
    data3 = ser.readline() #leitura dos dados
    list3 = list(data3) #conversão dos dados para um array
    global UTC #declaração da variável
    global in_checksum3 #declaração da variável
    global out_checksum3 #declaração da variável
    UTC = bytes(list3[3:9]) #criação de array de bytes
    in_checksum3 = hex(xor_checksum_string(bytes(list3[0:9])))
    out_checksum3 = list3[10] #cálculo de checksum da
    #mensagem de latitude
    #e gravação em variável do checksum vindo do ESP32
```

```

#-----#
def SOG():
    #função de leitura dos dados de velocidade
    ser.write('4'.encode()) #transmissão código de pedido/comando
    data4 = ser.readline() #leitura dos dados
    list4 = list(data4) #conversão dos dados para um array
    global SOG #declaração da variável
    global in_checksum4 #declaração da variável
    global out_checksum4 #declaração da variável
    SOG = bytes(list4[3:9]) #criação de array de bytes
    in_checksum4 = hex(xor_checksum_string(bytes(list4[0:9])))
    out_checksum4 = list4[10] #cálculo de checksum da
    #mensagem de latitude
    #e gravação em variável do checksum vindo do ESP32
#-----#
def COG():
    #função de leitura dos dados de rumo
    ser.write('5'.encode()) #transmissão código de pedido/comando
    data5 = ser.readline() #leitura dos dados
    list5 = list(data5) #conversão dos dados para um array
    global COG #declaração da variável
    global in_checksum5 #declaração da variável
    global out_checksum5 #declaração da variável
    COG = bytes(list5[3:9]) #criação de array de bytes
    in_checksum5 = hex(xor_checksum_string(bytes(list5[0:9])))
    out_checksum5 = list5[10] #cálculo de checksum da
    #mensagem de latitude
    #e gravação em variável do checksum vindo do ESP32
#-----#
def heading():
    #função de leitura dos dados de proa
    ser.write('6'.encode()) #transmissão código de pedido/comando
    data6 = ser.readline() #leitura dos dados
    list6 = list(data6) #conversão dos dados para um array
    global heading #declaração da variável
    global in_checksum6 #declaração da variável
    global out_checksum6 #declaração da variável
    heading = bytes(list6[3:9]) #criação de array de bytes
    in_checksum6 = hex(xor_checksum_string(bytes(list6[0:9])))
    out_checksum6 = list6[10] #cálculo de checksum da
    #mensagem de latitude
    #e gravação em variável do checksum vindo do ESP32
#-----#
def serial_reboot():
    #reinicialização da porta serie
    ser.close() #para eliminar qualquer mensagem de erro
    ser.open() #caso contrario o RPI entra em loop
#-----#
def pop_sensor_info():
    #função de pedido de todos os
    #dados dos sensores com reinicialização
    #entre cada pedido
    serial_reboot()
    latitude()
    time.sleep(0.1)
    serial_reboot()
    longitude()
    time.sleep(0.1)
    serial_reboot()
    UTC()
    time.sleep(0.1)
    serial_reboot()
    SOG()
    time.sleep(0.1)
    serial_reboot()

```



```

COG()
time.sleep(0.1)
serial_reboot()
heading()
time.sleep(0.1)
#-----#
def processed_info:                                     #função de verificação da
pop_sensor_info()                                     #integralidade das mensagens
if in_checksum1 == out_checksum1:                     #comparando os checksums
    final_latitude = str(latitude)                   #recebidos e os calculados
    print(final_latitude)                             #caso contrário é impressa
else:                                                  #mensagem de erro
    print ('Pedido de Latitude falhou')
if in_checksum2 == out_checksum2:
    final_longitude = str(longitude)
    print(final_longitude)
else:
    print ('Pedido de Longitude falhou')
if in_checksum3 == out_checksum3:
    final_UTC = str(UTC)
    print(final_UTC)
else:
    print ('Pedido de Unidade de Tempo falhou')
if in_checksum4 == out_checksum4:
    final_SOG = str(SOG)
    print(final_SOG)
else:
    print ('Pedido de Velocidade de Superfície falhou')
if in_checksum5 == out_checksum5:
    final_COG = str(COG)
    print(final_COG)
else:
    print ('Pedido de Rumo falhou')
if in_checksum6 == out_checksum6:
    final_heading = str(heading)
    print(final_heading)
else:
    print ('Pedido de Proa falhou')
#-----#
while True:
    processed_info()                                   #rotina do RPI
    time.sleep(1)                                     #inicialmente faz pedido de todos os
    ser.write('8'.encode())                           #dados sensoriais e finalmente faz
    time.sleep(1)                                     #pedido de transmissão pelos canais
    ser.write('7'.encode())                           #de comunicações das mensagens
    time.sleep(1)                                     #pré-definidas:
    ser.write('9'.encode())                           #-> 7,9 e 10 (frases NMEA por LoRa)
    time.sleep(1)                                     #-> 8 (dados por WiFi)
    ser.write('10'.encode())
#-----#

```


Apêndice D – Plataforma de Testes (433 MHz)

```
// LoRa -> ESP32 //
```

```
/*-----*/  
//BIBLIOTECAS//  
#include <SPI.h> //protocolo SPI  
#include <LoRa.h> //comms chip SX1276 - LoRa (433MHz)  
/*-----*/  
#define MASTER //Esta configuração permite a utilização de  
//um programa principal (comum aos dois ESP)  
//onde são definidos as funções em comum  
//e depois permite ligar a um programa  
//secundário onde estão as funções  
//especificas de cada ESP, para isso basta  
//comentar esta linha se quisermos carregar  
//o firmware para o ESP do VENT-SUP ou não  
//comentar se quisermos carregar o firmware  
//para o ESP da estação em terra  
/*-----*/  
#define SCK 5  
#define MISO 19  
#define MOSI 27 //pins de ligação SPI SX1276  
#define SS 18  
#define RST 14  
#define DI00 26  
#define BAND 433E6 //banda de frequência -> 433 MHz  
/*-----*/  
const String GETDATA = "getdata"; //frases identificativas de  
const String SETDATA = "setdata"; //pedido de transmissão de msg  
/*-----*/  
void setupLoRa() { //Inicialização do sx1276  
    SPI.begin(SCK, MISO, MOSI, SS); //modulo LoRa  
    LoRa.setPins(SS, RST, DI00);  
    if (!LoRa.begin(BAND, true)) {  
        Serial.println("Erro ao inicializar o LoRa!");  
    } //Configuração dos parâmetros LoRa:  
    LoRa.enableCrc();  
    //LoRa.setTxPower(20); // [2,20]  
    //LoRa.setSpreadingFactor(12); // [7,12]  
    //LoRa.setSignalBandwidth(62.5E3); // [250E3,125E3,62.5E3]  
    //LoRa.setCodingRate4(codingRateDenominator); // [5,8]  
    LoRa.receive(); //ao iniciar o sx1276 este é sempre  
} //colocado em modo de receção  
/*-----*/
```


Apêndice E - Plataforma de Testes (433 MHz – Estação em Terra)

```
// LoRa -> Estação em terra -> ESP32 //
```

```
/*-----*/
#ifdef MASTER          //configuração que define firmware secundário
                        //do ESP da estação em terra
/*-----*/
#define INTERVAL 1000          //declaração de variáveis
long lastSendTime = 0;
/*-----*/
void setup() {              //inicialização do monitor série
    Serial.begin(115200);    //e módulo LoRa
    setupLoRa();            //(função no firmware principal)
    LoRa.setTxPower(20);    //potência de transmissão = 20 dBm
}
/*-----*/
void loop() {               //rotina principal da estação em terra
    if (millis() - lastSendTime > INTERVAL) {
        lastSendTime = millis();    //condição de verificação
        send();                    //que permite que não haja
    }                               //colisão de pacotes
    receive();
}
/*-----*/
void send() {               //função que requisita a receção dos dados
    LoRa.beginPacket();    //dos sensores
    LoRa.print(GETDATA);
    LoRa.endPacket();
}
/*-----*/
void receive() {            //função de receção dos dados
    int packetSize = LoRa.parsePacket();
    if (packetSize > SETDATA.length()) {    //se o tamanho dos dados
        String received = "";              //recebidos for maior que a
        while (LoRa.available()) { //mensagem de resposta pre-definida
            received += (char) LoRa.read(); //é porque recebemos novos
        }                                   //dados, sendo estes guardados
        int index = received.indexOf(SETDATA); //numa string
        if (index >= 0) {    //se a mensagem SETDATA estiver presente na
            //msg será mostrado no monitor série todos os dados exceto essa msg
            String data = received.substring(SETDATA.length());
            String waiting = String(millis() - lastSendTime);
            Serial.println();    //calcula o tempo de ida e volta da msg
            Serial.println(data);
            Serial.println("Tempo: " + waiting + "ms");
            Serial.println("RSSI: " + String(LoRa.packetRssi()));
            Serial.println("SNR: " + String(LoRa.packetSnr()));
        }    //apresentação dos dados e relação sinal ruído e
    }        //intensidade do sinal de cada pacote
}
#endif
/*-----*/
```


Apêndice F - Plataforma de Testes (433 MHz – VENT-SUP-EN)

```
// LoRa -> VENT-SUP-EN -> ESP32 //
```

```
/*-----*/
#ifndef MASTER          //configuração que define firmware secundário
                        //do ESP da estação em terra
*-----*/

//BIBLIOTECAS//
#include <SPI.h>         //protocolo SPI
#include <LoRa.h>        //comms chip SX1276 - LoRa (433MHz)
#include <TinyGPS++.h>    //modulo GPS - ublox M8N
#include <Wire.h>        //protocolo I2C
#include <Adafruit_Sensor.h> //camada data-logging para sensores
#include <Adafruit_BNO055.h> //modulo IMU - BOSCH BNO055
#include <utility/imuMaths.h> //cálculos da fusao sensorial
*-----*/

TinyGPSPlus gps;
Adafruit_BNO055 bno = Adafruit_BNO055(55);
*-----*/

int count = 0;
float r = M_PI / 180.0f;
float d = 180.0f / M_PI;
float magd = -72.4;
float rawHeading;
float heading;
double hdt;
float latitude;
float longitude;
int latitudeD;
float latitudeDM;
int longitudeD;
float longitudeDM;
int longitudeD0;
float longitudeDM;           //declaração de variaveis
int utc;
String equador;
String meridiano;
int satellites;
float sog;
float cog;
int ddate;
int hora;
int minutos;
int segundos;
double distanceToGS;
*-----*/

void setup() {
  Serial.begin(115200);      //inicialização do monitor
  série
  setupLoRa();              //e módulo LoRa(função no firmware principal)
  LoRa.setTxPower(20);      //potência de transmissão = 20 dBm
  Serial1.begin(9600, SERIAL_8N1, 12, 15);
  bno.begin();              //inicialização do IMU
}
```

```

    bno.setExtCrystalUse(true);
}
*-----*/
void loop() {
    int packetSize = LoRa.parsePacket();

    if (packetSize == GETDATA.length()) {
        String received = "";
        while (LoRa.available()) {
            received += (char) LoRa.read();
        }

        if (received.equals(GETDATA)) {
            String data = readData();
            Serial.println("Criando pacote para envio");
            LoRa.beginPacket();
            LoRa.print(SETDATA + data);
            LoRa.endPacket();
            Serial.println("Enviou: " + String(data));
        }
    }
}
*-----*/
static void smartDelay(unsigned long ms)
{
    unsigned long start = millis(); //código de verificação de
    do                               //posição de GPS atualizada
    {
        while (Serial1.available())
            gps.encode(Serial1.read());
    } while (millis() - start < ms);
}
*-----*/
void readBNO() { //função de leitura dos dados do IMU
    sensors_event_t event;
    bno.getEvent(&event);
    rawHeading = (event.orientation.x); //leitura da orientação
    heading = rawHeading - 90;          //no eixo xx (horizontal)
    if (rawHeading < 90 && rawHeading >= -179.99) {
        heading = rawHeading + 270;
    }
    hdt = heading - magd;                //conversão da orientação obtida
    if (hdt >= 360) {                    //pelo IMU em leituras de proa
        hdt = hdt - 360;                //simulando uma bussola compensada
    }                                   //por valores do giroscopio
    if (hdt < 0) {
        hdt = hdt + 360;
    }
}
*-----*/
void readGPS() {
    latitude = gps.location.lat();
    if (latitude < 0) { //função de leitura dos
        latitude = latitude * -1; //dados de GPS
    }
    longitude = gps.location.lng();
    if (longitude < 0) {
        longitude = longitude * -1;
    }
}

```



```

}

latitudeD = floor(latitude);
latitudeDM = ((latitude - latitudeD) * 60);

longitudeD = floor(longitude);
longitudeD0 = longitudeD * 10;
longitudeDM = ((longitude - longitudeD) * 60);

utc = (gps.time.value()) / 100;
equador = (gps.location.rawLng().negative ? "N" : "S");
meridiano = (gps.location.rawLng().negative ? "W" : "E");
satelites = (gps.satellites.value());
sog = (gps.speed.knots());
cog = (gps.course.deg());
ddate = (gps.date.value());
hora = (gps.time.hour());
minutos = (gps.time.minute());
segundos = (gps.time.second());
static const double GS_LAT = 38.6631, GS_LON = -9.146906;
distanceToGS =
    TinyGPSPlus::distanceBetween(
        gps.location.lat(),
        gps.location.lng(),
        GS_LAT,
        GS_LON);
if (millis() > 5000 && gps.charsProcessed() < 10)
    Serial.println(F("No GPS data received: check wiring"));
smartDelay(100);
}
*-----*/
//função onde se faz a leitura dos dados a enviar, tendo como valor
//de saída a frase NMEA de telemetria e distância a estação em terra
String readData() {
    readGPS();
    readBNO();
    return String("$GPGGA," + String(utc) + "," + String(latitudeD) +
String(latitudeDM, 5) + "," + String(equador) + "," +
String(longitudeD0) + String(longitudeDM, 5) + "," + String(meridiano)
+ ",1," + String(satelites) + "\n") + ("GPRMC," + String(utc) + ",A,"
+ String(latitudeD) + String(latitudeDM, 5) + "," + String(equador) +
"," + String(longitudeD0) + String(longitudeDM, 5) + "," +
String(meridiano) + "," + String(sog) + "," + String(cog) + "," +
String(ddate) + "\n") + ("$HCHDT," + String(hdt) + "\n") + ("VENT-SUP
encontra-se a: " + String(distanceToGS, 2) + " m"));
}
#endif
*-----*/

```


Apêndice G - Plataforma de Testes (2.4 GHz – VENT-SUP-EN)

```
// 2.4 GHz -> VENT-SUP-EN -> ESP32 //

/*-----*/
//BIBLIOTECAS//
#include <WiFi.h> //protocolo comms WiFi 802.11 (2.4GHz)
#include <ESP32WebServer.h> //webserver
#include "FS.h" //manipulação de ficheiros cartão SD
#include "CSS.h" //template de webserver
#include <SD.h> //funções básicas cartão SD
#include <SPI.h> //protocolo SPI
/*-----*/

ESP32WebServer server(80);
SPIClass spiSD(HSPI);
/*-----*/

#define SD_CS 13 //declaração de variáveis
#define SDSPEED 27000000 //velocidade máxima permitida
String webpage = "";
bool SD_present = false;
/*-----*/

void setup(void) {
    Serial.begin(115200); //inicialização de ligação serie
    spiSD.begin(14, 2, 25, 13); //ligação ao SD card reader
    WiFi.begin("VG_INTRANET", ""); //ligação à rede criada pelo router
}

Serial.println("\nConectado a " + WiFi.SSID() + " Endereço IP: " +
WiFi.localIP().toString());
if (!SD.begin(SD_CS, spiSD, SDSPEED)) {
    Serial.println(F("Cartão SD não está inserido corretamente"));
    SD_present = false;
} //inicialização e codigos de
else //erro dos equipamentos
{
    SD_present = true;
}

//comandos de inicialização do webserver//
server.on("/", HomePage);
server.on("/download", File_Download);
server.on("/upload", File_Upload);
server.on("/fupload", HTTP_POST, []() {
    server.send(200);
}, handleFileUpload);
server.on("/stream", File_Stream);
server.on("/delete", File_Delete);
server.on("/dir", SD_dir);
//fim dos request commands//
server.begin();
Serial.println("Webserver pronto a utilizar");
}
/*-----*/

void loop(void) { //rotina do ESP32
    server.handleClient(); //função de gestão dos clientes
}
```

```

} //ligados à rede invocando funções a pedido
/*-----*/
void HomePage() { //função de criação de página principal do
    SendHTML_Header(); //webserver para gestão de ficheiros do cartão
    append_page_footer(); //chamadas funções para criação
    SendHTML_Content(); //dos componentes da página web
    SendHTML_Stop();
}
/*-----*/
void File_Download() { //função download de ficheiros
    if (server.args() > 0 ) { //introdução válida do nome do ficheiro
        if (server.hasArg("download")) SD_file_download(server.arg(0));
        //chamada da função de download do
        //cartão ocorre se pressionado enter, que redireciona
        //para o endereço 192168.0.150 com o argumento introduzido
    } else SelectInput("Enter filename to download", "download",
"download");
}
/*-----*/
void SD_file_download(String filename) {
    if (SD_present) { //função de download do cartão
        File download = SD.open("/") + filename);
        if (download) { //download do ficheiro
            server.sendHeader("Content-Type", "text/text");
            server.sendHeader("Content-Disposition", "attachment; filename="
+ filename);
            server.sendHeader("Connection", "close");
            server.streamFile(download, "application/octet-stream");
            download.close();
        }
    }
}
/*-----*/
void SendHTML_Header() {
    server.sendHeader("Cache-Control", "no-cache, no-store, must-
revalidate");
    server.sendHeader("Pragma", "no-cache");
    server.sendHeader("Expires", "-1");
    server.setContentLength(CONTENT_LENGTH_UNKNOWN);
    server.send(200, "text/html", "");
    append_page_header();
    server.sendContent(webpage); //funções que recorrem ao
    webpage = ""; //template da biblioteca CSS.h
} //para criar o cabeçalho, copo e
void SendHTML_Content() { //rodapé da página web
    server.sendContent(webpage);
    webpage = "";
}
/*-----*/
void SendHTML_Stop() { //função de término de
    server.sendContent(""); //tarefa e disrupção de comunicação
    server.client().stop(); //com o cliente
}
/*-----*/
void SelectInput(String heading1, String command, String
arg_calling_name) {
    SendHTML_Header();
    webpage += F("<h3>"); webpage += heading1 + "</h3>";
}

```

```

    webpage += F("<FORM action='/'"); webpage += command + "'
method='post'>"; // Must match the calling argument e.g. '/chart'
calls '/chart' after selection but with arguments!
    webpage += F("<input type='text' name=''"); webpage +=
arg_calling_name; webpage += F("' value=''><br>");
    webpage += F("<type='submit' name=''"); webpage += arg_calling_name;
webpage += F("' value=''><br><br>");
    append_page_footer();
    SendHTML_Content();           //função que permite a entrada
    SendHTML_Stop();             //de texto para selecionar o ficheiro
}                                //para efetuar o download
/*-----*/

```